

ADAPTIVE AND INEXACT APPROACHES FOR
ENERGY-EFFICIENT AND VARIATION-AWARE NANOMETER VLSI DESIGN

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Jaeyoon Kim

January 2014

© 2014 Jaeyoon Kim

ADAPTIVE AND INEXACT APPROACHES FOR ENERGY-EFFICIENT AND VARIATION-AWARE NANOMETER VLSI DESIGN

Jaeyoon Kim, Ph. D.

Cornell University 2014

Adaptive circuit design technique and error-tolerant computing have both been suggested as potential methodologies for addressing two major hurdles facing the future of semiconductors: increasing variability and decreasing energy-efficiency, both of which becoming especially prominent as transistor scaling becomes increasingly aggressive with gate lengths down to sub-20nm and below. Adaptive circuit design partially relaxes the operating safety margins by dynamically adjusting system parameters such as supply voltage, body bias, and operating frequency; however, it cannot fully eliminate such margins since it must guarantee computational correctness in all cases including the worst-case combinations of extreme variations. Error-tolerant computing such as error detection/correction or resilient hardware has been proposed to relax these margins. While some of the potential benefits of error-tolerant computing have been revealed, their implementation requires a significant amount of design, power, and complexity overhead. This dissertation presents a novel methodology to relax some of the design tradeoffs present in current adaptive circuit design techniques by employing a double-gate MOSFET (DGMOSFET) device as the main circuit element, and introduces a more efficient error-tolerant computing framework, which will hereby be referred as “*Inexact Computing*” in this dissertation.

This dissertation presents the implementation of adaptive circuit design techniques using an independently-biased back-gated DGMOSFET, the details of which includes the theory of the DGMOSFET device modeling, new design techniques for compensating parametric variations, and achieving better energy-efficiency and noise robustness. Threshold voltage tuning using back-gate of the DGMOSFET was compared with a conventional body-bias method. This technique is a promising solution to control the transistor's threshold voltage while reducing undesirable effects at the sub-50 nm device technology nodes. An automatic adaptive circuit for threshold voltage tuning was implemented using DGMOSFET devices in 45nm CMOS technology. Simulation results show that this circuit compensates for static and dynamic variations. This adaptation approach using DGMOSFETs along with adaptive supply voltage scaling allows simultaneous optimization of power and performance according to application-specific workload and requirements. Simulation results using a 45nm CMOS technology indicate that this adaptive circuit design can provide 50% higher performance for the same energy, or consume 40% less energy for the same performance. In contrast to conventional methods which only employ dynamic voltage scaling, adaptive tuning of threshold voltages reduces power consumption while maintaining high noise margin.

As another solution for mitigating variability and power issues, this dissertation also introduces the theoretical framework for probabilistic circuit representations of conventional CMOS digital logic and reveals the relationship between the error probabilities vs. energy. Using probabilistic modeling in sub-50nm silicon transistor technology, the relationship between statistical uncertainties and errors are elucidated

for different configurations and topologies and design the trade-offs are quantified. Gate-level implementation of the probabilistic CMOS logic is validated by circuit simulations of a commercial 45nm SOI CMOS process technology. Presenting as an example a practical ALU architecture where voltages can be scaled from most significant to least significant bit blocks, the potential benefits of this technique are shown. A calculation error of 10^{-6} , an error rate quite tolerable for many computational tasks, is shown to be possible with a total power reduction of more than 40%. More importantly, the relation of error probabilities and energy from our probabilistic approach follows the second law of thermodynamics, regardless of scale or topology of a circuit.

Finally, this dissertation verifies the suggested relationship of error vs. energy by a prototype image signal processing system implemented on an FPGA. The processing of a 2D RGB color image using this prototype is used to verify this relationship. For each R, G, and B color component, 2D 3-tap FIR image filters are implemented using hard IP of the FPGA. Measurements were performed using programmable pulse generators and a logic analyzer to minimize the dependency on FPGA synthesis and place/route design flows. Subsequent experiments demonstrate the feasibility of using inexact computing for specific error-tolerant applications such as human vision. An image processing error of 1.2×10^{-6} is shown to provide acceptable image quality while reducing the total power consumption by 30%.

BIOGRAPHICAL SKETCH

Jaeyoon Kim was born on April 10, 1974 in Incheon, Korea to Tae-sup Kim and Nanjoo Kang. Jaeyoon lived in Incheon before attending Gyeong-gi Science High School in Suwon, Korea. After two years of high school, he attended the Korea Advanced Institute of Science and Technology in Daejeon, Korea and obtained his Bachelor of Science in Electrical Engineering in 1997. Following completion of his undergraduate degree, Jaeyoon started his career at a startup company in Korea as an alternative to military service. He became the youngest CTO in 2003 at the age of 29. He led the first Korean group to commercialize a CNC Wire-cut Electrical Discharging Machine and also developed various machine tools and automated manufacturing applications for automotive tires, nuclear reactor fuel rods and fixtures, LCD panels, and semiconductor packaging. In May 2007, he joined the Nanoscale ElectroScience Research Group at Cornell University under the supervision of Prof. Sandip Tiwari, where he worked on multi-gate FETs using UTSOI, error-tolerant and probabilistic circuit designs, wireless inter-tier interconnects in 3D ICs, advanced FPGA architecture using future memory devices, and ultra-low power design methodology using *Inexact Computing*. He is currently working as a circuit design engineer in the Foundation IP group at Qualcomm in San Diego, CA working on high-speed cache of next generation LTE MODEM and low-power circuit architectures for application processors. He obtained his Doctor of Philosophy in Electrical and Computer Engineering in December 2012. Jaeyoon married Eunhee Han in 2004 and is now the father of an adorable 5-year-old girl, Seohyun Sophia Kim.

To my parents, Tae-sup Kim and Nanjoo Kang

And to my wife, Eunhee Han

ACKNOWLEDGEMENTS

My journey to a doctoral degree had many twists and turns. When I decided to embark on this journey, I was fascinated to an advanced research but did not have any detailed plan or roadmap. My courage, or even foolhardiness, would not have finally transformed into fruition without helps and encouragements from many people. First and foremost, I would like to thank my advisor, Prof. Sandip Tiwari, for his direction and guidance that kept my research focused and steady. While he and I do not always agree on certain topics, he has shown unwavering supports in completing this journey.

I would also like to thank Prof. Rajit Manohar for his advice and support as a member of my graduate committee. I am particularly thankful for his generosity to allow my participation in his weekly group meeting to discuss about VLSI designs, access to his design resources such as EDA tools and design evaluation boards, and the administrative help during my medical leave of absence in 2008. I would also like to thank the other member of my committee, Prof. Edwin Kan for his time and valuable suggestions. I also thank Prof. Alyssa Apsel for the Talisman server during my entire years of graduate study at Cornell. I offer many thanks to Prof. Ehsan Afshari for his encouragement and consideration during the last year of my study. A significant portion of my research work has been done through collaboration with Dr. Paul Solomon at the IBM T. J. Watson. His advice and expertise on device modeling have elevated my work to even higher level. I also owe my gratitude to Dr. Solomon for his initial work on the DGMOSFET modeling.

I would like to express my sincere gratitude to the past and present members of our research group. Dr. Hao Lin, Bilal Khan, Joshua Rubin, and Eric Yu. I am particularly indebted to Dr. Moonkyung Kim for his mentorship during my transition from M.Eng to Ph.D. program. Also, many thanks go to Dr. Brian Bryce and Weimin Chan for interesting discussions on various non-engineering topics. The AVLSI group members also have my deepest gratitude for their friendship, in particular Dr. Song Peng, Dr. David Fang, Ben Hill, Rob Karmazin, and Jonathan Tse. I would particularly like to thank Dr. Filipp Akopyan and Carlos Tadeo Ortega for sharing lab equipment and their knowledge in terms of VLSI designs, as well as helpful discussions. I am also thankful to Dr. Xiao Wang and Dr. Rajeev Dokania for enjoyable discussions on circuits and Cadence tools.

I would like to thank other Korean colleagues and friends for their friendship and care: Dr. Eugene Hwang, Changhyuk Lee, Sunwoo Lee, Sungyun Park, Jeeho Ryoo, Dr. Jungheyon Hwang, Dongue Lee, Dr. Youngchul Choi, Dr. Seungkeun Yoon, Jinsup Kim, and Dr. Jaikyung Jung. Among them, I would especially like to thank Dr. Eugene Hwang for spending many nights in helping my first tapeout, lots of proof readings, and his warmhearted care for my family. Also, special thanks to Changhyuk Lee for his friendship and assistance, Sunwoo Lee for his humor, and Sungyun Park for sharing unforgettable moments of the last winter at Cornell. I am also thankful to Yanning Li for his friendship during the first year in Ithaca.

Other friends during our stay in Ithaca include Prof. Inhwan Han and his family, Dr. Chungeun Lee and his family, Don Karr, Chi-heon Yi, James Orcutt, and Dr. Hyungsoon Park. I am also grateful for their friendship and encouragements. Other

friends and colleagues in San Diego are Suna Jong and her family, Steve Kim, Jongwon Lee, Hoon Ryu, Richard Song, and Anil Kota. I would like to thank all of them for their help to settle down in San Diego and Qualcomm. I also want to thank Hanseung Kim for his friendship and invaluable discussions.

I am extremely grateful to my previous advisor, Prof. Choongki Kim for motivating me to embark this journey. Also my two best friends Seongab Kim and Kwansuhk Oh have my biggest gratitude for their long-lasting friendship. Also, this work would not be possible without the support of National Science Foundation and NYSATR through CNS. I am also grateful for the sponsorship of DARPA through MIT Lincoln Laboratory's 2nd and 3rd MPW 3D-IC tapeout.

Without the encouragement and support of my father, Tae-sup Kim, I would not be able to even think of this journey. I thank my mother, Nanjoo Kang for her love and trust. I thank my brother, Jaewoo Kim, for his support and taking care of our entire family while I was in the U.S. I am also grateful to my sister, Jaemin Kim, for her love and thoughtfulness. I would also like to extend my thanks to my parents-in-law, Pyungwoong Han and Boon-nam Lim for helping us after my daughter's birth, and my sister-in-laws, Hyekyung and Hoeyoung Han for their love and support during my hardship at Cornell.

Last but not least, I thank my wife Eunhee Han for her love and support. She made a tremendous sacrifice in taking care of the family, and helped me to stay focused. My daughter, Seohyun Sophia Kim, has always been the source of my motivation to arrive at this very important milestone and beyond.

TABLE OF CONTENTS

Biographical Sketch.....	iii
Dedication.....	iv
Acknowledgements	v
Table of Contents	viii
List of Figures.....	xi
List of Tables	xiii
1 Introduction	1
1.1 The End of CMOS Scaling and the Emergence of Mobile Computing	1
1.2 Research Scopes and Dissertation Outlines	10
2 Adaptive Circuit Design Using Independently Biased Back-Gated Double-Gate MOSFETS.....	13
2.1 Motivation and Background	13
2.2 Double-Gate MOSFET and its Modeling for Simulations.....	15
2.2.1 Physics-Based Device Model and Comparison with 2D Numerical Simulation.....	16
2.2.2 Compact Device Modeling for Circuit Simulations	22
2.3 Body-Biasing versus Independently Biased DGMOSFET	28
2.3.1 Reverse Body Biasing	28
2.3.2 Forward Body Biasing.....	28
2.3.3 Independently biased Back-Gate	30
2.4 Parametric Variation Compensation using Adaptive Circuits.....	32
2.4.1 Design of Adaptive Circuit.....	32
2.4.2 Simulation Results.....	37
2.5 Power-Performance Adaptation	40
2.6 Adaptive Circuit Design for Improved Noise-Margin	48
2.7 Chapter Summary	55
3 Inexact Computing using Probabilistic Circuits	58
3.1 Motivation and Background	58
3.2 Probabilistic Approach for Non-deterministic CMOS Logic.....	61
3.2.1 CMOS Logic Implementation using Probabilistic Approach	62
3.2.2 Characterization of Probabilistic Behavior of CMOS Inverter	63

3.3	Simulation Framework and Experimental Methodology	66
3.4	Impact of Input Noise on the Probability of Error	69
3.5	Error-Energy Relationship for Gate-Level Logic Implementation	76
3.6	Power Savings via Inexact Computing.....	80
3.6.1	MSB-LSB Weighted Scaling of Supply voltages	80
3.6.2	Architecture of Adder.....	81
3.6.3	Simulation Results.....	85
3.6.4	Ultra Low-power Data-path circuit Design Methodology using Probabilistic Circuit.....	87
3.7	Chapter Summary	93
4	Ultra-Low Power ALU and DSP core for Inexact Computing	95
4.1	Motivation and Background.....	95
4.2	ALU Design for MSB-LSB weighted supply voltage scaling	98
4.2.1	Adder Design.....	98
4.2.2	Multiplier Design.....	101
4.2.3	Multiplier-Accumulator for Digital Signal Processing	109
4.3	Image Processing Example: Inexact Computing.....	110
4.3.1	Experiment and Measurement scheme	111
4.3.2	Experiment using Manufacturer-supplied Design Platform.....	116
4.3.3	Experiment using Conventional FPGA Design Flow	119
4.3.4	Experiment using Minimal Hardware Implementation.....	126
4.4	Measurement Result and Discussion.....	131
4.5	Chapter Summary	134
5	Future Research Directions and Conclusions.....	135
5.1	Double-Gate MOSFETs and its adaptive design applications	135
5.2	Statistical simulation framework using Probabilistic CMOS.....	139
5.3	Inexact computing: MSB-LSB weighted scaling scheme	143
5.4	Inexact computing: Other application examples	143
5.5	Conclusions	147
	Appendix	149
	A Probabilistic approach for Statistical representation of delay variations.....	149
	B Probabilistic methodology for Statistical variations: Simulation details and how to	156
	Bibliography.....	158

LIST OF FIGURES

1.1	The End of Traditional CMOS Scaling	2
1.2	Battery capacity power consumption indexes with the maximum output level in cellular transmitters	4
1.3	Variability-Induced Failure Rates for Three Canonical Circuit Types	6
1.4	Power Supply-Dependent Failure Rates for Three Canonical Circuit Types.....	8
2.1	A generic schematic cross-section of planar double-gate MOSFET showing definition of terms.....	16
2.2	Visualizations of potential along the channel of a DGMOSFET showing the current paths	18
2.3	Electron concentration distribution perpendicular to the oxide interface for 10 nm body thickness of symmetric DGMOSFET structure..	19
2.4	Drain current of the same DGMOSFET structure.....	20
2.5	Threshold voltage shift vs. gate length	21
2.6	Comparisons of mixed-mode model with 2-D numerical simulations.....	22
2.7	Simulated characteristics. I_d vs. V_{fg} with different V_{bg}	23
2.8	I_d vs. V_{ds} with different V_{bg}	24
2.9	Capacitances vs. V_{fg} with $V_{bg} = 0.2$ V (Upper). L_g is 50 nm	27
2.10	Capacitances vs. V_{ds} with $V_{bg} = -0.2$ V, $V_{fg} = 1$ V (Lower). L_g is 50 nm	27
2.11	Propagation delay of 12 inverter chain vs. independently biased back-gate voltage and body-bias	29
2.12	Comparison of threshold voltage tunability	29
2.13	Modeling procedure for Verilog-A coding and Program flow control for mixed mode DGFET model.	31
2.14	Scheme of Adaptive Circuit	34
2.15	Schematic of Delay Pattern Generator with 2x, 4x, 8x, and 16x reference clock	35
2.16	Schematic diagram of Delay Monitoring block	35
2.17	Error Detector Timing Chart in case of speed up	36
2.18	Functional components for Signal Processor and Back-bias generating Block.....	37
2.19	Critical path propagation delay is reduced to satisfy design specification by applying back-gate bias	39
2.20	Normalized delay in terms of threshold voltage with different supply voltage values	41
2.21	Comparison of V_{th} adaptation and V_{dd} scaling	43
2.22	Summary of operating modes for optimized V_{dd} and V_{th}	45
2.23	Schematic diagram of circuit design for V_{dd}/V_{th} adaptation	45
2.24	Adaptation of V_{dd} and V_{th}	46
2.25	Noise margin of a unit-sized inverter	50
2.26	Power vs. NM and Power vs. Critical path delay of multiplier blocks	52

2.27	Simulated inverter switching threshold voltage versus PMOS-to NMOS width ratio.....	54
2.28	Inverter circuit for noise filtering, designed using 45nm DGMOSFETs and SOI SGMOSFETs	54
2.29	Output signal comparisons	55
3.1	Output signal of the CMOS inverter and output signal with probability of errors.....	63
3.2	Noise coupling of an inverter and noise distribution analysis.....	65
3.3	A comparison method for evaluating the probability of error and a simulation schematic for a CMOS inverter	69
3.4	Models of various types of statistical variations in CMOS inverter	71
3.5	Probability of error vs. noise rms values with various types of noise sources, input-coupled only, output-coupled only, input/output-coupled, V_{dd} -coupled, and GND-coupled.	72
3.6	Probability of error vs. noise rms values	72
3.7	AC gain and frequency response plot of unit-sized inverter	73
3.8	Noise amplification effect at different biases	74
3.9	Input noise translates into jitter during input transition.....	75
3.10	Simulation scheme for 1-bit full-adder circuit.	76
3.11	Relationship of energy per bit operation vs. probability of error for inverter, NAND, and XOR.....	79
3.12	Errors in MSB position produce larger calculation errors than errors in LSB.....	80
3.13	Implementation of MSB-LSB weighted scaling of supply voltages for 32-bit CCS-CSS adder.	83
3.14	Logic implementation of 4-bit Conditional Carry Select adder block and 4-bit Conditional Sum Select adder block	84
3.15	MSB-LSB bit selection map for voltage scaling.....	86
3.16	Energy vs. calculation error for 32-bit adder by applying MSB-LSB weighted scheme	86
3.17	Probability of error for a 4-bit CCS-CSS adder block.	89
3.18	Power vs. Calculation Error	89
3.19	Power vs. Calculation Error for the different numbers of independent voltage sources	92
4.1	Logic Implementation of 8-bit Conditional Carry Select adder block	99
4.2	Energy vs. calculation error for 64-bit adder by applying MSB-LSB weighted scheme	100
4.3	Multiplication operation example.....	101
4.4	Partial product generation logic.....	101
4.5	An area-optimized circuit implementation for modified Radix-4 Booth encoder	104

4.6	A schematic of circuit implementation for modified Radix-4 Booth selector.....	105
4.7	Transmission-gates 4:2 compressor	106
4.8	Application of MSB-LSB weighted scaling scheme at the stage of column sum for the case of 16-bit multiplier	108
4.9	Energy vs. calculation error for 32-bit multiplier by applying MSB-LSB weighted scheme	108
4.10	MAC implementation for $Y = A \cdot B + C$	109
4.11	The mechanics of image filtering with $N \times N = 3 \times 3$ filter	114
4.12	3×3 image filter for sharpness enhancement and original image and processed image after applying the above 3×3 image filter	115
4.13	Base Reference Design Block Diagrams.....	117
4.14	Modification to the Base Reference Design Block to accommodate the desired measurement goal	118
4.15	Another modification to get rid of Ethernet software dependency.	119
4.16	Test platform implemented by Virtex-6	120
4.17	Image processing block implemented at 150MHz	122
4.18	Supply voltages vs. image processing error	123
4.19	Modified scheme for increasing clocks to BlockRAMs and image processing block	124
4.20	Frequency vs. Error rate for the increased operating clock of 250MHz	125
4.21	A new method to minimize the software dependency of an FPGA implementation flow.....	125
4.22	Detailed experiment set-up for minimal hardware implementation, which minimize the software dependency of the conventional FPGA tool flows	127
4.23	Implementation of 2-D 3 tap FIR filter using three 1-D 3 tap FIR filters	128
4.24	Signal flow diagram for 1-D 3 tap FIR filter.....	128
4.25	Implementation of the above signal flow using the Virtex-6's DSP48E1 macro	128
4.26	Overall implementation flow for 2-D 3 tap image filter for image sharpening.....	129
4.27	Relationship between Power vs. Calculation Error.....	130
4.28	Image quality is significantly degraded and may not acceptable	131
4.29	Image quality degradation is noticeable	132
4.30	Quality degradation is hardly discoverable	132
4.31	Image with no error	133
5.1	An example of fabrication of the planar type back-gated MOSFET.....	135
5.2	Transformation of the device structure from Tri-gate FinFET to Fin-type Double-Gate MOSFET.....	136
5.3	Layout of FinFET	137
5.4	Circuit schematic for selective use of DGMOSFET to minimize the layout overhead	137
5.5	Butterfly plots and read margins extraction results	138

5.6	Statistical models are provided by foundries for functional robustness under an influence of variability.....	140
5.7	Comparison between Monte Carlo simulation and Probabilistic circuit approach	142
5.8	Hybrid video encoder	144
5.9	Circuit schematic for the sensor to detect a probability of error	147
A.1	A clock tree of five stages clock buffers	150
A.2	Simplified switch model of dynamic behavior of CMOS inverter.....	152
A.3	Delay distribution function from the 10k of Monte Carlo runs.....	152
A.4	Delay distribution function from the probabilistic approach	153
A.5	$1/\sqrt{\text{delay}}$ fitting normal plot for the 10k Monte Carlo runs.....	154
A.6	$1/\sqrt{\text{delay}}$ fitting normal plot for the probabilistic simulation	155
B.1	Random noise signal to mimic the situation of the traditional Monte Carlo method	157

LIST OF TABLES

3.1	Simulation parameters for Probabilistic CMOS logic circuits	68
4.1	Radix-4 modified Booth encoding values and the corresponding Boolean expressions	102
4.2	Simulation result for both of circuit implementations.....	103
4.3	Simulation result for various circuit implementation styles for Booth Selector	103
5.1	Failure criteria, which determines how many sigmas are required for the target design.....	140
5.2	Comparison between Monte Carlo simulation and Probabilistic circuit approach	141

CHAPTER 1

INTRODUCTION

1.1 The End of CMOS Scaling and the Emergence of Mobile Computing

Within the past few years, it has become apparent that it is no longer sustainable for the semiconductor industry to continue keeping pace with Moore's Law. Even advocates of this law announced that the CMOS scaling reached its limit as shown in Figure 1.1. Chip manufacturers have relied on continued dimensional downscaling to achieve exponential growth in transistor count per die, but the performance enhancements due to simply shrinking the dimensions of the planar transistors have already plateaued and even dimensional downscaling itself will be ending soon.

Two main impediments stand in its way: increasing device variability and power consumption. As device dimensions shrink, their variability and failure rates increases, presenting challenges to the traditional deterministic paradigm of digital circuit design where correct computation must be ensured. Additionally, in recent years, quantum mechanical effects have begun to appear in various forms, the most deleterious example of which is the increase in leakage current. This leakage current serves no useful purpose and accounts for an increasingly larger portion of the total device power consumption. This ultimately reduces the strong square law dependence of power consumption on supply voltage. In order to limit this leakage current, the threshold voltage cannot be scaled as aggressively as the device dimensions. Both of these considerations act to limit the effectiveness of supply voltage scaling, a very

useful knob in reducing the total power consumption when dynamic switching power is dominant. Coupled with the fact that parasitics – which is not scale as strongly with device dimensions – are increasingly becoming a larger part of the total device switching capacitance, this all means that each device is becoming increasingly inefficient, even as they are getting smaller. Ultimately, these consequences negate the benefits that made scaling such a formidable force to begin with and continuing this trend is clearly unsustainable.

With the inception of the pervasive ubiquitous computing era of the last decade, handheld wireless devices such as mobile phones have become one of the most prolific electronic devices. As a consequence, reducing power consumption in portable device is becoming a top priority for semiconductor manufacturers. As ever increasing features and exploding demand for performance continue to be integrated into these products, there is an ongoing need to develop innovative ways to reduce power consumption and extend battery life.

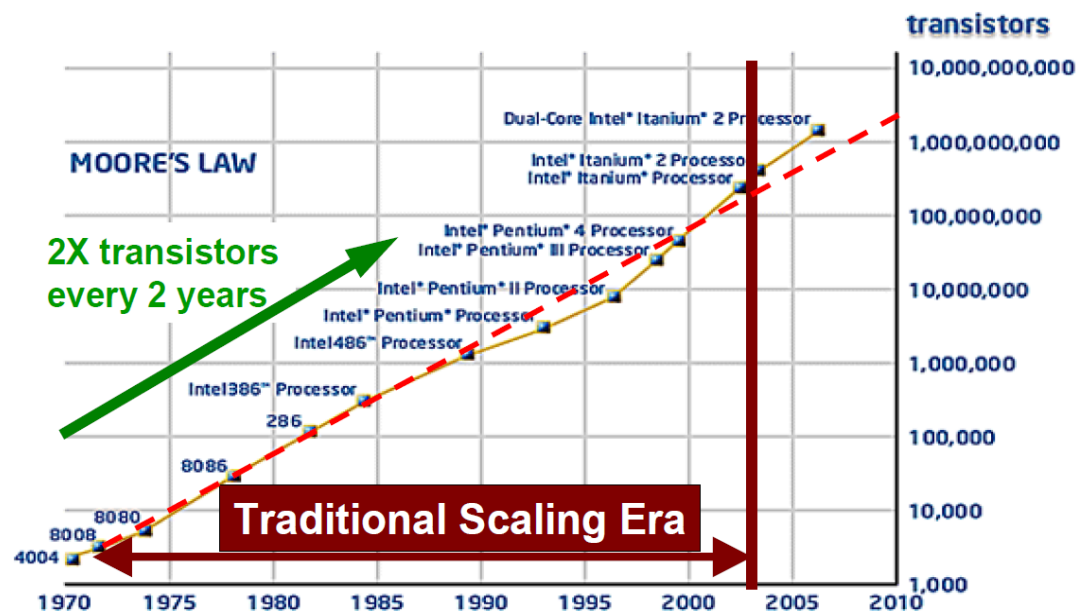


Figure 1.1 The End of Traditional Scaling Era [1].

In a battery-operated device, the available energy is limited, and the rate of power consumption determines the time between recharges. Size, aspect ratio, and weight of batteries typically are set by the application, e.g., different battery types are each used for smartphones, tablets, and laptops. The allowable battery size of a smartphone would be at most 10-15 cm³, as dictated by prevailing smartphone in the current market. Given a particular battery technology, the expected operation time of device in between recharges – for example, cell phone users today would expect multiple day of standby time and 7 to 8 hours of talking time – set an upper bound on the power dissipation for the different operational modes. This in turn limits what functionality can be supported by the device, unless breakthroughs in low-power design methodology can be achieved. For instance, the average power dissipation limit of a cell phone is approximately 3W, constrained by today's battery technologies. This determines whether your phone will be able to support digital video streaming, MP3 audio, and 3G/LTE network support or WiFi connectivity. As already observed in Figure 1.2, battery capacity doubles approximately every 10 years [2]. This represents an improvement of 3-7% every year and lags significantly behind the device feature size downscaling trend. Because of this significant lag, the increasing power consumption in scaled devices has emerged as a large impediment to integrating greater functionality in these mobile devices.

In addition to limiting the total functionality that can be integrated into a single chip, the increasing power density of these devices is starting to place limits on transistor integration density due to thermal limitations. As more transistors that are increasingly inefficient get packed into smaller areas, the heat generated is starting to

reach the practical limits of cooling. Furthermore, with the advent of System-On-a-Chip (SoC) in mobile devices, diverse functionalities integrated in close proximity with different workloads and activity profiles result in a creation of hot spots and abrupt temperature gradients over the die. Hot spots may impact the long-term reliability of the chip. Also, temperature gradients complicate the chip verification due to the varying propagation delay over the functional blocks since circuit performances strongly depend on temperature. Techniques such as reducing the gradients or the power density of the systems to mitigate the packaging problems are becoming prohibitively expensive to be used in consumer electronics or mobile products. Increasing power density impedes taking full advantage of device downscaling with respect to performance, reliability, and verification.

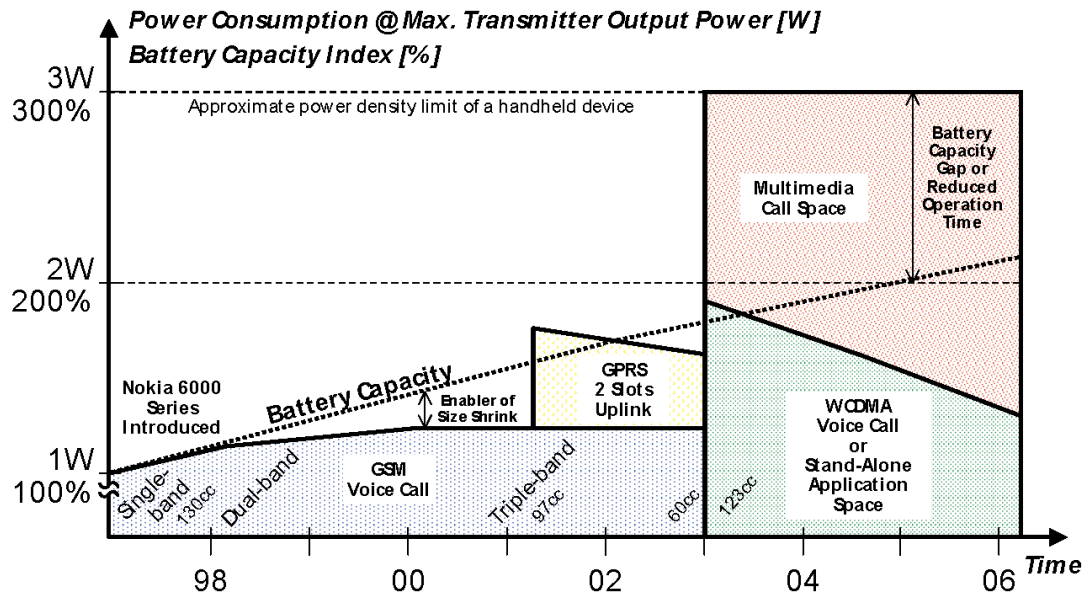


Figure 1.2 Battery capacity and power consumption indexes with the maximum output power level in cellular transmitters

Another important aspect of scaling down the transistors is the decreased ability to handle fabrication process variations. As transistors and passive components become

smaller, fewer atoms make up the individual parts. For example, the gate oxide in the 28nm SiON process node is only about 5 atoms thick [3]. In this case, a difference of only a single atomic layer provide up to 20% variation of device parameters. This large and unpredictable process variation significantly complicates the design process, and requires at least 20% of safety operating margins to guarantee proper functionality. Process variations can be due to Random Dopant Fluctuations (RDF) including well proximity effect (WPE) [4], line-edge roughness arising from the limits of lithographic technology and so on. Besides these static variations which do not change over time once determined during fabrication, transistor characteristics can also change over the lifetime of the integrated circuit as a result of hot carrier effects or negative bias temperature induced (NBTI) changes in the threshold voltage. In addition, new techniques to improve transistor performance by using strain engineering [5] also will bring additional sources of variation.

To make predictions about the impact of these types of phenomena on future technologies, the ITRS [6] has selected three commonly used, basic CMOS logic circuits to use as a baseline for comparison: an SRAM bitcell, a simple latch, and the common inverter. This follows the precedent of using design-oriented circuit to identify future trends. It is common to have many millions of SRAM bits, and millions of latches and inverters in current high-performance processors.

Failure probabilities for the three basic circuits in future high-performance (HP) technology nodes were obtained by simulating their behavior under the influence of manufacturing process variability. The simulations used the Predictive Technology Model (PTM) with variability estimates for both general logic and SRAM circuits for

bulk CMOS technologies down to the 16 nm node. The respective criteria for failure were as follows.

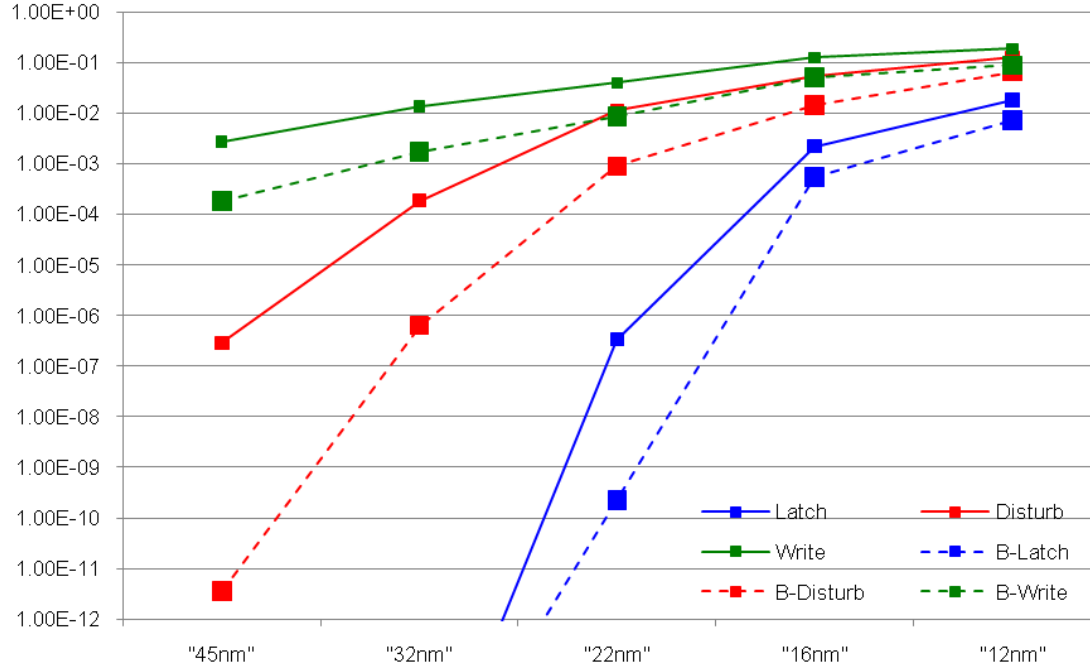


Figure 1.3 Variability-Induced Failure Rates for Three Canonical Circuit Types, ITRS 2011

1. For the SRAM bitcell, two distinct failure modes were explored: (a) the writability fail, where the SRAM is unable to store one of the two Boolean values, and (b) the read disturb fail, where the act of reading an SRAM causes the contents of the cell to reverse polarity.
2. For the latch, the CLK-to-Q (clock to output) delay was measured, with failure corresponding to CLK-to-Q delay 10 times its nominal value. Since the CLK-to-Q delay is an important part of the timing of any digital circuit, such a drastic increase is likely to cause timing failures similar to what would be observed if the latch were to experience a hard fault.

3. For the inverter, the pair delay (the delay of two inverters in series) was measured, with failure corresponding to pair delay 10 times its nominal value. Given the pervasive use of inverters as buffers on long interconnect wires, such a drastic increase in delay is, again, likely to cause significant timing failures.

Failure rates for the three basic circuits are shown in Figure 1.4. The figure shows the technology nodes on the x-axis, and the failure probability on the y-axis. Three families of curves are shown for the latch and the two SRAM failure modes as indicated above. The inverters do not show failure rates above the minimum value of the plot at one part per billion. Each family has two curves, one solid and one dashed; the dashed line denotes the performance of the circuit when the device widths are scaled up by a factor of 40%. Some conclusions can be reached from these simulation results as listed below:

1. SRAM failure rates, already a significant problem that requires extensive design intervention such as the introduction of massive redundancy circuitry, will continue to be a problem and will require even more circuit (e.g., R/W assist circuits [7] and architectural innovations to combat increasing manufacturing variability).
2. Latches, which share some broad similarity with SRAM but were more robust at the 45nm node, reach SRAM failure rates at or around the 22nm node. This will necessitate the introduction of new circuit and/or architectural techniques to ensure correct operation.
3. Enlarging circuits (reverse scaling) is only moderately effective at controlling the impact of variability.

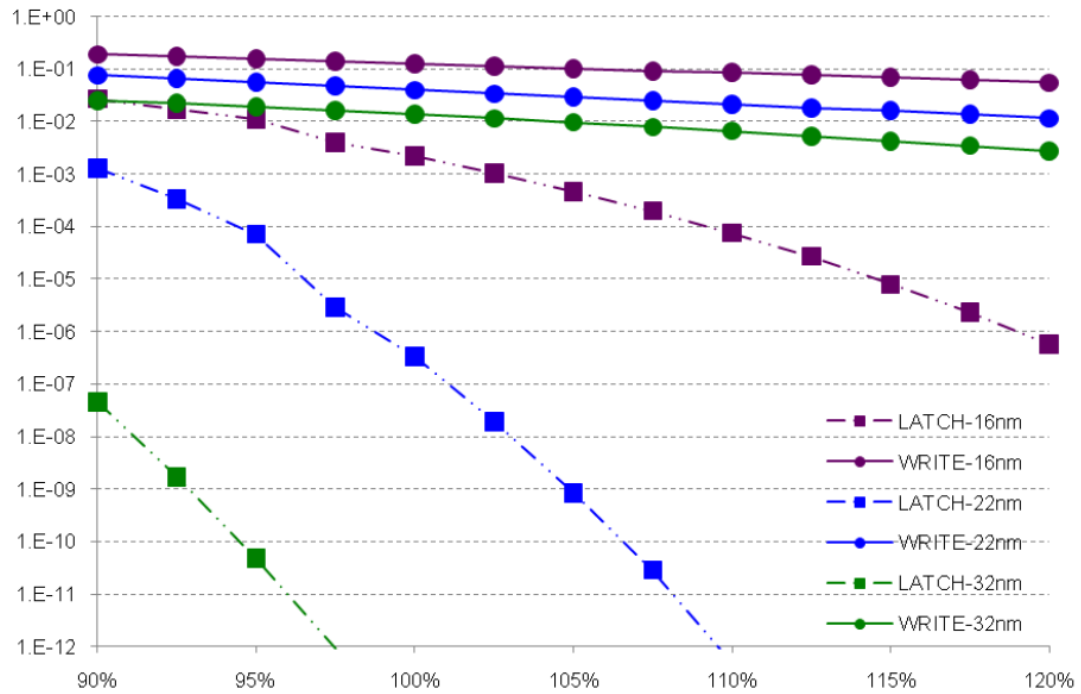


Figure 1.4 Power Supply-Dependent Failure Rates for Three Canonical Circuit Types, ITRS 2011

An important lever in reducing the impact of variability is the power supply. It is well appreciated that raising the power supply voltage can significantly reduce circuit failure rates due to variability. Of course, this comes at the expense of additional power consumption, which is already a major factor for all types of designs. In order to understand the impact of power supply voltage on failure rates, simulations reported in Figure 1.4 was extended to include the dependence of failure rates on power supply voltage. The range shown was from 10% below the nominal power supply voltage, a common worst-case assumption in digital circuit design, to 20% above the nominal power supply voltage. Results are shown in Figure 1.5, where the x-axis is the power supply (as a percentage of the nominal supply for each technology node), and the y-axis is the probability of failure. Only the results for the latch and the SRAM write

failure at the 32nm, 22nm and 16nm nodes are plotted for simplicity. We can see that the power supply has a very strong impact on latch failure rates, and a somewhat more modest impact on SRAM Write failure rates. The improvement is still about one order of magnitude for this range of voltage variations. This observation leads to a clear engineering tradeoff between power and robustness. Absent other sources of innovation at the device and circuit levels, one of the few effective levers for reducing circuit fail rates is power. Given that power itself is now one of the major design drivers, designers and technologists will have to take great care in developing balanced solutions for this problem.

In summary, this section introduces the reason why most of the researchers in the semiconductor area claim that we have entered an era of power-limited scaling. This means power considerations are primary factors determining how process, transistor, and interconnecting parameters are scaled. Furthermore, based on the current state and outlook of a number of emerging technologies (e.g., carbon nanotubes transistors, graphene, molecular electronics, etc.) to replace deep-submicron CMOS, it seems improbable for the near future that we will see a “revolutionary” transition to a new device technology that would significantly relax this tradeoff, such as that seen during the introduction of the bipolar transistor to replace vacuum tube technology or the introduction of the MOSFET to in turn replace that. Ultimately, for the time being, it will be new design approaches and innovative low-power computing architectures that will propel technology forward.

1.2 Research Scope and Dissertation Outlines

Based on the previous discussion of the current needs for continuing advancement in semiconductor with an awareness of the power and variability constraint, this dissertation suggests three strategies initiated from each of the different hierarchies in integrated circuits, i.e. device, circuit, and system-level hierarchies. The three different hierarchical approaches can be summarized as follows:

1. Device-level approach: as a promising candidate for reducing Short-Channel Effect (SCE) and process variability, the Double-Gate MOSFET (DGMOSFET) device is employed. Its compact device model is created using Verilog-A for high-throughput VLSI circuit simulation.
2. Circuit-level approach: adaptive circuit design using independently biased back-gated DGMOSFET devices is proposed to address the previous problems in traditional DVFS and body-biasing schemes. Also, probabilistic circuit approaches for digital CMOS logic are suggested and the fundamental framework using this concept for circuit simulation is established.
3. System-level approach: the *Inexact Computing* using the probabilistic CMOS methodologies suggested at the circuit level is proposed and investigated for ultra-low power computing. Several System-level implementations such as ALU, DSP, and image processing system are implemented and verified by FPGA implementation as holding promise for future ultra-low power designs.

The remainder of this dissertation is focused on elucidating these different hierarchical approaches. Chapter 2 starts by introducing the basic concept of the DGMOSFET and exploring the viable options for implementing a more efficient compact device model for fast and robust VLSI circuit simulation. After comparing traditional adaptive circuit design techniques using DVFS and body-bias with our proposed methodologies, adaptive circuit designs using DGMOSFET as a circuit element are shown for compensating parametric variations, further optimizing the tradeoffs between powers vs. performance, and enhancing noise immunity.

We explore and develop the details of a probabilistic approach in the context of device modeling and circuit design in Chapter 3. A new circuit-level characterization and simulation methodology is demonstrated using EDA tools with IBM's 45 nm 12SOI process technology. Beginning with the simplest form of gate-logic, the inverter, as the basic building block for development of probabilistic CMOS logic, the methodology is applied to the higher-levels of integration, culminating in a specialized CCS-CSS (Conditional Carry Select – Conditional Sum Select) adder architecture.

In chapter 4, by employing the methodologies suggested in the previous chapter, 64-bit adder, 32-bit multiplier, and 32-bit MAC (Multiplier-Accumulator) designs are demonstrated. In addition, an example of image processing implemented on FPGA is presented to verify our *Inexact computing* methodology. A detailed explanation of systematic experiment is highlighted and simulation results are discussed.

Finally, we conclude by discussing some future research directions based on the work presented in this dissertation. The advantage for integration of DGMOSFET into a commercial CMOS process beyond 10nm is identified and remaining challenges as

well as potential new applications are discussed. The benefits and potential future directions of the presented probabilistic circuit approach are highlighted together with the potential applications for ultra-low power and high-performance computing such as many-core architecture and Near-Threshold Computing (NTC). The advantages of the circuit techniques and the probabilistic design paradigm presented in this work suggest that they may be promising candidates for relaxing some of the stringent power-performance tradeoffs presented in deep submicron CMOS technologies and help driving continued development of the next-generation computing.

CHAPTER 2

ADAPTIVE CIRCUIT DESIGN USING INDEPENDENTLY BIASED BACK-GATED DOUBLE-GATE

2.1 Motivation and Background

Advancements in device size enable improvement in computing performance while decreasing the cost per die. However, size reduction also results in undesired effects of increasing variability including those arising from static process variations and changes during use such as transistor degradation and aging. Due to nonscalability of the threshold voltage and underlying limits on the sub-threshold slope, supply voltage scaling has slowed or plateaued in recent years while maintaining technology scaling trend. Since manufacturing variability has a critical influence on device characteristics, supply voltage plateauing is largely influenced by the need to preserve operating margins. As a consequence, the supply voltage in modern technologies is significantly higher than originally suggested by the constant field scaling theory [8], which has directly led to dramatic increase in power density. Today, in digital design, power places severe constraints on technology scaling and dissipation levels are often raised to the practical limit of cooling [9]. Adaptive circuit design techniques have recently been introduced to overcome these unwanted by-products of device shrinkage [10]. Adaptation can compensate for process variations and respond to power budget, various workloads, and environments. A basic adaptive circuit design technique is the

dynamic adjustment of the supply voltage, also known as dynamic voltage scaling (DVS). While this technique is widely used in modern microprocessor designs, it presents a rather inflexible tradeoff between power consumption and circuit performance since it does not have a full control over the leakage power. Specifically, lowering V_{dd} can greatly lower the total dynamic power consumption, but it also results in significant performance degradation and reduced robustness due to the shrinking of noise margins. In modeling of high performance bulk MOSFET design, under the constraint of bounded performance, it has been suggested that the optimal ratio of leakage power to total power consumption is approximately 30% [11]. While dynamic power consumption shows a strong dependence on the supply voltage, the dependence of leakage power on supply voltage is not as strong. Therefore, simply reducing the supply voltage makes it hard to meet this optimum ratio. Using a dynamic body-biasing method, many research groups have suggested threshold voltage control as another approach to dynamically adjust leakage power [12, 13]. However, this technique becomes less effective with technology scaling because Short-Channel Effects (SCE) dominate over the effect of body-biasing in today's submicron CMOS processes [14].

In this chapter, an independently biased Double-Gate MOSFET (DGMOSFET) is employed to investigate an alternative approach of adjusting the threshold voltage. DGMOSFET has been known to have improved immunity against SCE, approximately a factor of two improvements over its single-gate counterpart [15]. In addition to this superior control over SCE, the tuning range and responsiveness of adjusting the threshold voltage with independently biased DGMOSFET is better than

conventional body-biasing methods. By simultaneously applying bias to the back-gate of DGMOSFET and adjusting the supply voltage, a more efficient adaptation technique is possible for compensating variations, partially relaxing the strong trade-offs between power consumption and performance, and increasing the noise margin. Evaluating DGMOSFET adaptation capability is the focus of this work. DGMOSFET with back and front-gate self-alignment has been demonstrated [17, 18], thus all natural advantages of CMOS processes are potentially available to this approach. In section 2.2, we describe a basic structure and characteristics of DGMOSFET, and introduce both physics-based as well as compact device models of DGMOSFET for 2-D numerical and circuit simulations. In section 2.3, circuit simulation is summarized comparing the effectiveness of threshold voltage control using independently biased DGMOSFET and conventional body-biasing techniques. Three different adaptive circuit designs are proposed in section 2.4, 2.5, and 2.6 for compensating variations, optimizing power and performance trade-offs, and improving noise-margin. In section 2.7, we summarize the advantages of our proposed circuits and conclude.

2.2 Double-Gate MOSFET and its Modeling for Simulations

To avoid the problems encountered during the reduction of device feature size, alternatives to the standard MOSFET configuration continue to be investigated, and one of the alternatives is the DGMOSFET (see Figure 2.1), as suggested by Frank et al [15]. In the double-gate configuration, the passive substrate is replaced by an actively biased gate so that the channel is modulated from top and bottom at the same time.

The DGMOSFET is electrostatically more robust than the standard single-gated MOSFET since the bottom gate shields penetration of the field from the drain, reducing SCE [15]. In addition to the DGMOSFET's enhanced immunity against SCE, another important attribute is the ability of the transistor to be controlled by different voltages of the two gates [16]. This provides an additional degree of freedom allowing for more flexible circuit design over using single-gate or common-gate counterparts, a feature especially important for adaptive circuit design. Devices with these attributes have been demonstrated by a number of groups [17, 18] in self-aligned geometry.

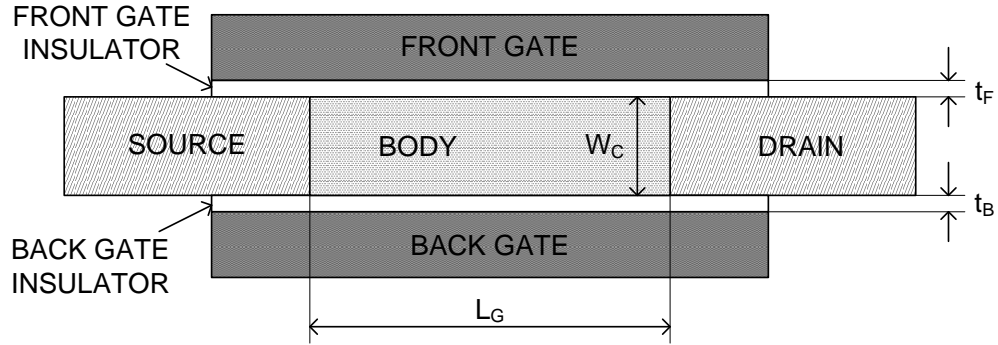


Figure 2.1 A generic schematic cross-section of planar double-gate MOSFET showing definition of terms

2.2.1 Physics-Based Device Model and Comparison with 2D Numerical Simulation

The simulations used a physics-based device model for the independently biased DGMOSFET, formulated by one of us [19]. It will be described in detail in this section. Our approach is to convert the DGMOSFET into an equivalent ‘elementary’

n-channel single gate MOSFET (SGMOSFET) to use a fairly conventional model to derive currents and capacitances for the SGMOSFET, and then re-apportion them to the terminals of the DGMOSFET. In the case of a DGMOSFET with a single channel, or in the case of symmetric-bias double gate, the conversion to this elementary form is straightforward. A more complicated situation arises in a MOSFET with two unequal channel biases as illustrated in Figure 2.2(b). In this case the potential developed in the channel, which increases toward the drain, causes the back channel (the weaker channel according to our arrangement) to pinch off before the front channel at a channel potential $V_c(x) = V_B$. While two channels are present, the potential separating them is only the self-confinement potential, which is small. As soon as the back channel pinch-off condition is exceeded, an electric field is generated between the two channels diverting the current from the back into the front channel. This situation, depicted in Figure 2.2(b), shows that the DGMOSFET under these conditions can be decomposed into two MOSFETs in series, one with a dual channel near the source with length, L_1 , and the other with single channel near the drain with length, L_2 as shown in Figure 2.2d. At first glance it may seem possible to represent the dual channel DGMOSFET as two MOSFETs in parallel because of the two parallel channels, but the strong electrostatic coupling between the two channels, especially in the case of a thin undoped body, precludes this.

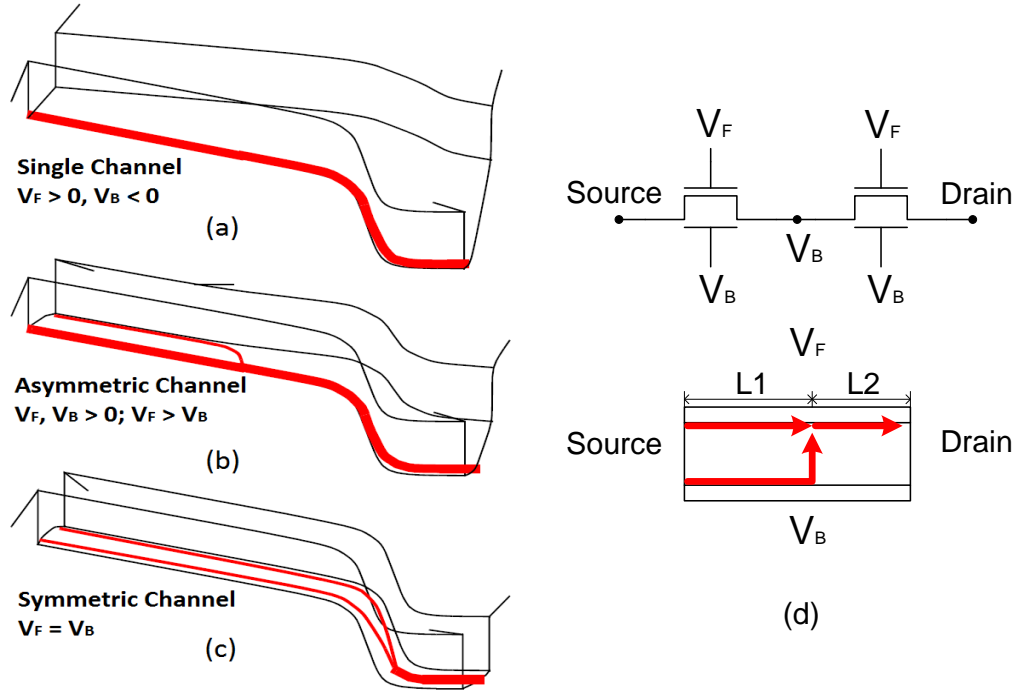


Figure 2.2 Visualizations of potential along the channel of a DGMOSFET showing the current paths: (a) under large negative back-gate and large positive front-gate bias, (b) under small positive back-gate and large positive front-gate bias, (c) under comparable positive front- and back-gate biases, (d) mixed-mode decomposition of a DGMOSFET with dual channels near the source into two separate MOSFETs and a single channel near the drain.

The “*mixed-mode*” scheme describes the DGMOSFET as two MOSFETs in series with the drain/source intermediate node being at potential V_B . The condition for *mixed-mode* operation is

$$V_F > V_B > 0 \text{ and } V_B < V_D.$$

This decomposition approach has physically been verified using the results of 2D device simulations. Figure 2.3 shows the concentration distribution of majority carriers in the channel with varying back-gate bias. The device is assumed to have a

symmetric double-gate structure with same gate work function for both front- and back-gate since the threshold voltage of a DGMOSFET with poly-silicon gate only is not suitable for normal operation [20]. For this symmetric configuration, the separate channel approach to evaluate total drain current has proved to be reasonable since two identical channels are formed with symmetric back-gate bias and the back channel is reduced and eventually merged to the front channel as the back-gate bias decreases to zero. In UTSOI devices, the geometrical quantum confinement of electrons spreads out from the Si-oxide interface into the Si slab and this tendency becomes more pronounced as the channel thickness decreases to less than 5 nm, below which merging of the front and back channels results [21, 22]

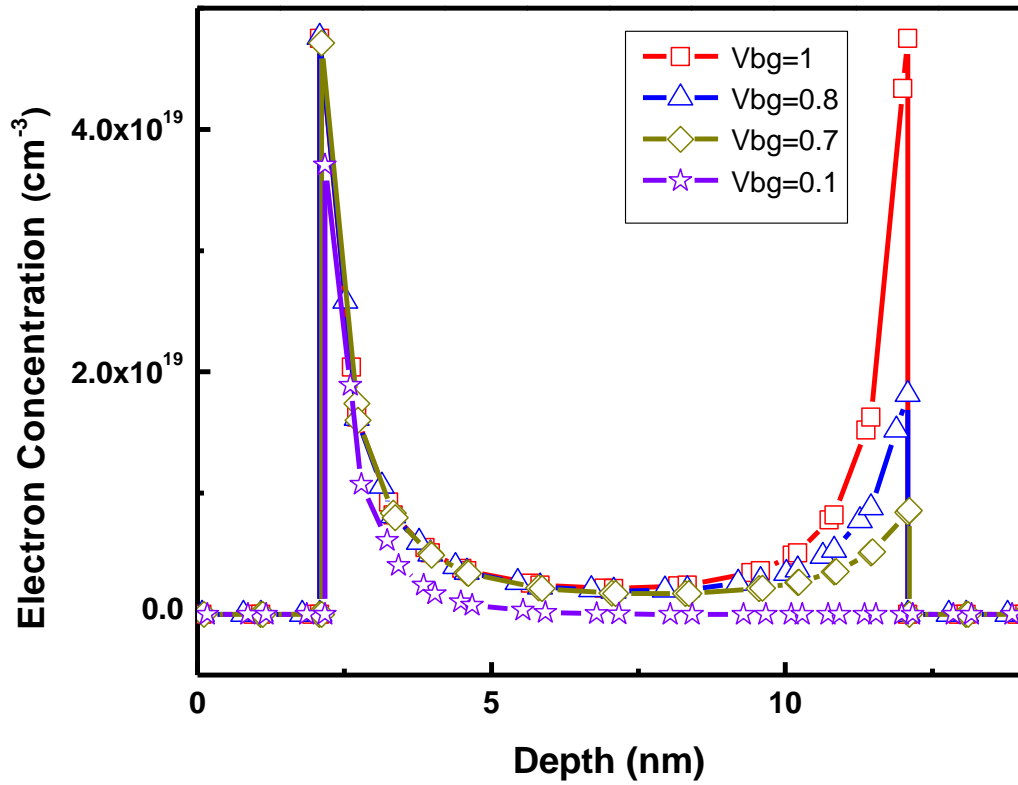


Figure 2.3 Electron concentration distribution perpendicular to the oxide interface for 10 nm body thickness of symmetric DGMOSFET structure. $L_g = 50$ nm.

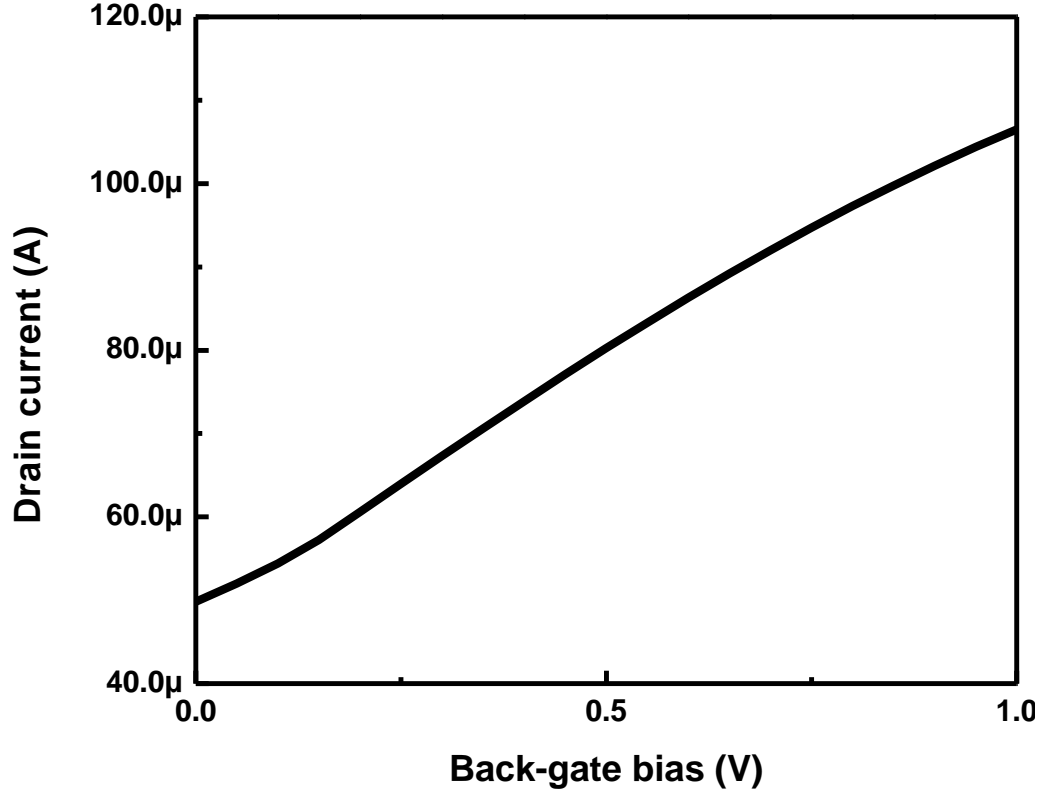


Figure 2.4 Drain current of the same DGMOSFET structure with $V_{fg} = 1.0V$ and $V_{ds} = 0.1V$

This merging is undesirable since it limits the tunability of the DGMOSFET. To avoid this, we use devices with channel thicknesses ranging from 5 to 10 nm where the dual-channel approximation is still valid [20]. Using this separate channel approach, the difficulties of rapidly calculating the drain current of DGMOSFET are simplified. Figure 2.4 depicts the increasing trend of drain current as back-gate bias increases with fixed front-gate and drain biases. This model was previously implemented in FORTRAN77 and ASX simulator and compared with two other works: (a) the DGMOSFET design space study of Wong et al. [23], and (b) a quantum-confined

channel DGMOSFET of Jeong et al. [20]. We have implemented this approach in Veriog-A compact device model with MATLAB M-file script as an intermediate form since MATLAB is a more convenient platform for debugging and verification. The equivalent MATLAB model is also compared with results from Atlas, a 2-D numerical device simulation tool [24]. Comparisons between our model and ATLAS are plotted in the Figure 2.7, Figure 2.8, Figure 2.9, and Figure 2.10 to show agreement.

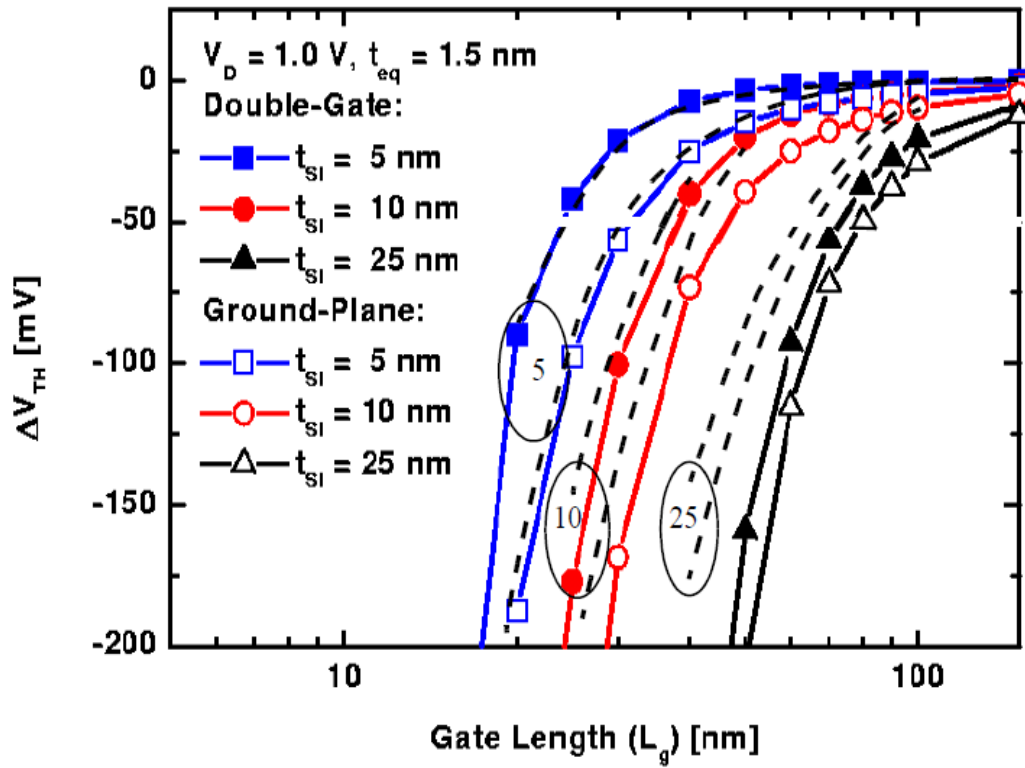


Figure 2.5 Threshold voltage shift vs. gate length, dashed line is the mixed-mode model and solid line is the data of Wong et al. [23]. Ground-Plane refers to a device where the back-gate is a ground plane.

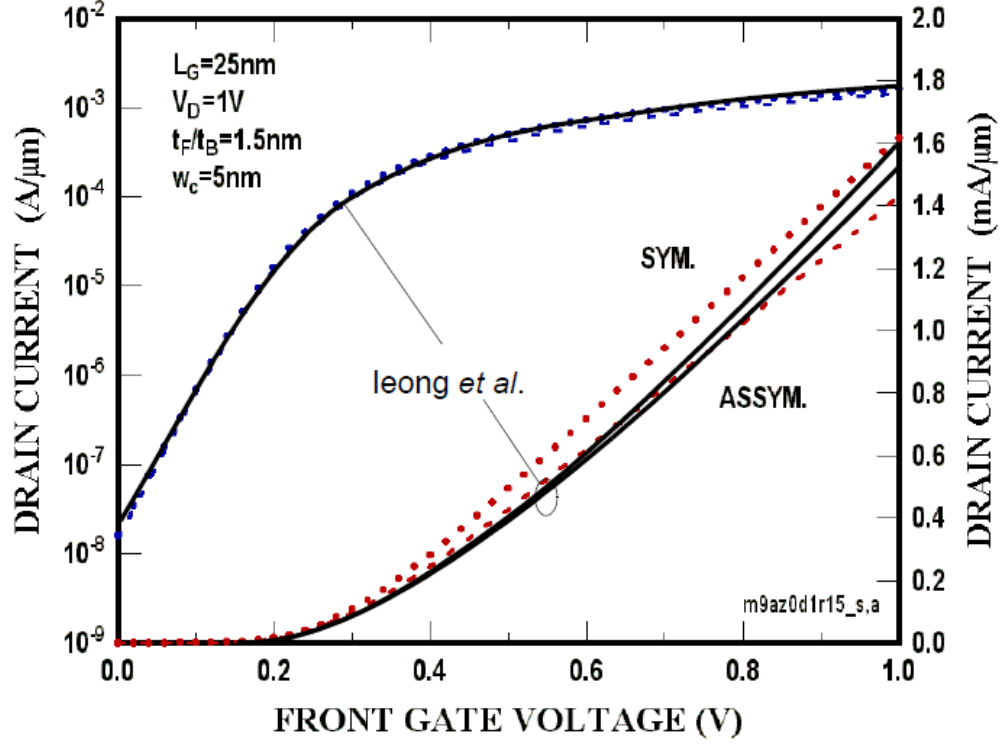


Figure 2.6 Comparisons of mixed-mode model with 2-D numerical simulations by leong et al. [20] (solid lines)

2.2.2 Compact Device Modeling for Circuit Simulations

Based on the physics-based device model, we developed a compact device model for circuit simulation. Since several iterative algorithms are needed for the solution of implicit equations describing the electrical behavior of the DGMOSFET, direct conversion of the physics-based model to conventional compact circuit model like

BSIM is difficult. Therefore, a robust and efficient analytical approximation for DGMOSFET device has been derived from the simplified physics-based model. This semi-empirical model uses fitting parameters to output the same current and capacitance values generated from the physics-based model. However, it still requires complicated expressions for the solution. Due to these difficulties, our compact device model can only simulate circuits containing a maximum of 1,000 DGMOSFET devices. The re-created compact device model is implemented in Verilog-A to be simulated using the Cadence Spectre and Synopsys HSPICE platform.

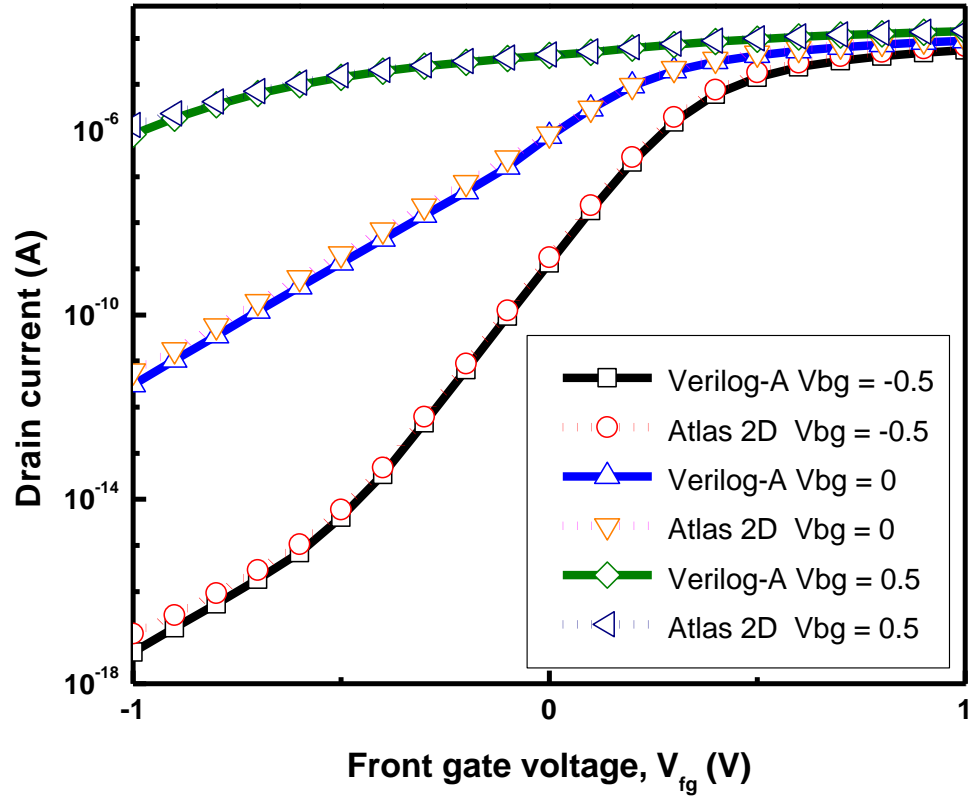


Figure 2.7 Simulated characteristics. I_d vs. V_{fg} with different V_{bg}

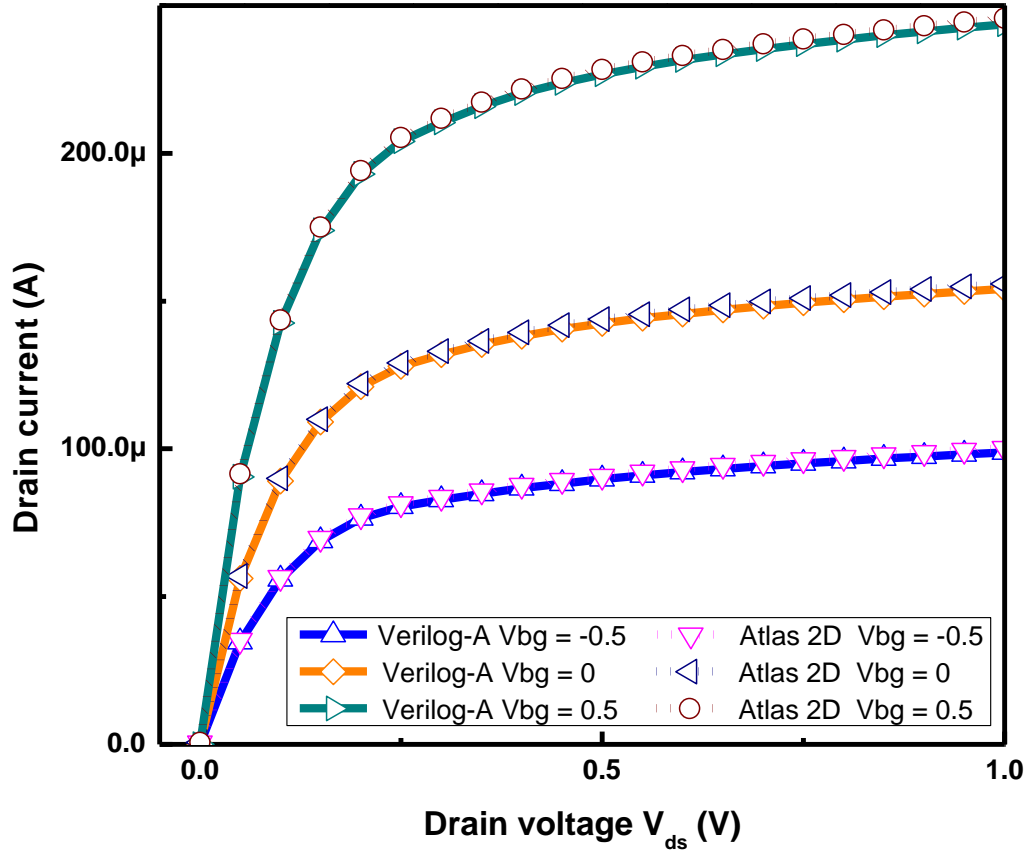


Figure 2.8 I_d vs. V_{ds} with different V_{bg} .

The physics-based device modeling algorithms introduced in Section 2A can be decomposed into two distinct internal routines. One is for calculating the currents and capacitances of SGMOSFET and the other is a mixed-mode algorithm for combining each portion of SGMOSFETs into a DGMOSFET depending on the independently biased back-gate voltages. However, direct conversion of above scheme to a Verilog-A model results in unmanageable difficulties such as convergence problems and long simulation times since our physics-based device model is not efficient as a compact

device circuit model. To increase robustness and simulation speed, a Verilog-A compact device model is re-created based on the following four criteria. First, a compact device model should not have its own iterative algorithm. A general SPICE-based circuit simulator like Spectre and HSPICE solves a KCL (Kirchoff's current law) equation at each node using the Newton-Raphson algorithm [25], which seeks the solution of a set of nonlinear equations through the iterative solutions of a sequence of linear equations. If the numerical algorithm implemented in the compact device model is iterative, then the number of iterations required for the final solution increases quadratically and we have no hope of expediency in the numerical convergence process. Second, a compact device model should be continuous throughout the entire region of operation without any discontinuities. In order to ensure the numerical robustness the derivatives of arbitrary order must be continuous at all voltage values of interest. The property is sometimes referred to as infinite-differentiability [26]. A general approach to guarantee this infinite-differentiability is to adopt a single equation to describe the drain current and terminal capacitances rather than with multi-region equations. This approach is the same as that used by the BSIM4 compact device models [27]. Third, another modification is needed for calculations of capacitance values. In Verilog-A, there is only one way to describe the capacitive behavior between two terminals, which is based on the equation: $i(t) = dQ/dt$. Since the capacitance values of our physics-based model is a function of voltages

$$i(t) = \frac{d}{dt} \{C(t) \cdot V(t)\} = C(t) \cdot \frac{d}{dt} V(t) + V(t) \cdot \frac{d}{dt} C(t)$$

by chain rule. However, Verilog-A does not support the use of the chain rule when

calculating time derivatives [28], so charges in the transistor structure must be modeled instead of capacitance. Last, we optimize our algorithm for the decomposition of a channel according to the voltages applied to the back-gate. Even though we developed a simplified compound mode to reduce the number of iteration for calculating the apportioned channel length and characterizing the transition behavior from single to dual channel, it still involves implicit equations and considerable number of iterations. Due to this incompleteness, the total number of DGMOSFETs is limited in actual circuit simulation. Using the proposed DGMOSFET Verilog-A model, we find that simulation speed and convergence is acceptable for circuits with up to 1,000 devices. The overall modeling procedures and flow control diagrams are summarized in Figure 2.13. Our Verilog-A code for DGMOSFET is embedded as a schematic symbol into the EDA tool environment. This model is implemented in the Spectre and HSPICE platforms. The simulated total propagation delays versus back-gate bias of two separate twelve inverter chains, one composed of only SGMOSFETs and the other composed of DGMOSFETs, are shown in Figure 2.11. The back-gate bias for the case of SGMOSFETs corresponds to the body bias of these devices.

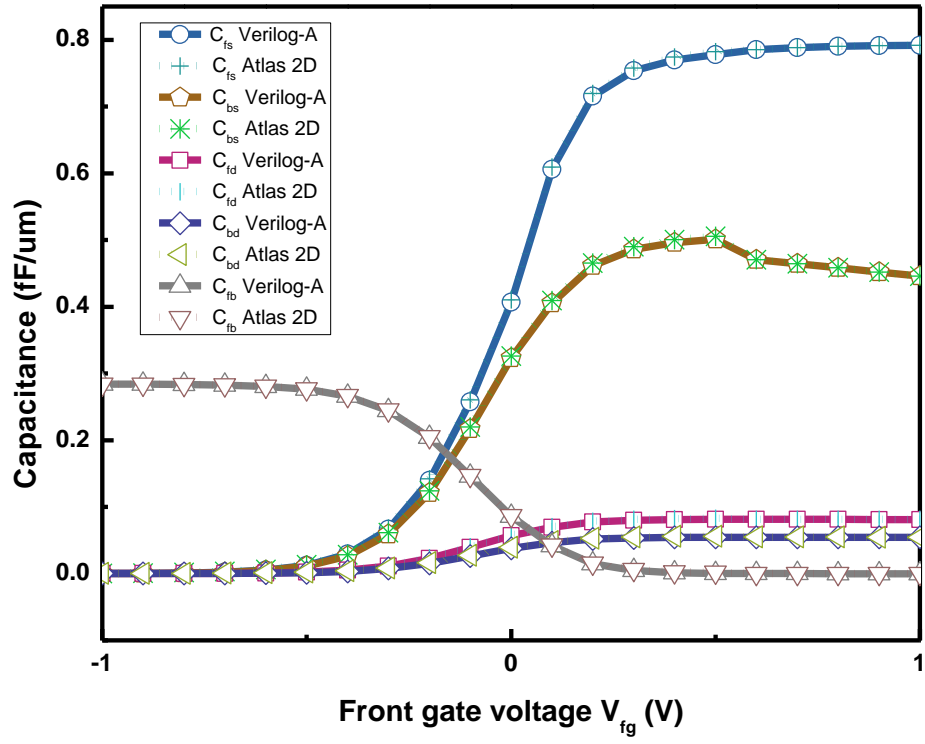


Figure 2.9. Capacitances vs. V_{fg} with $V_{bg} = 0.2$ V (Upper). L_g is 50 nm.

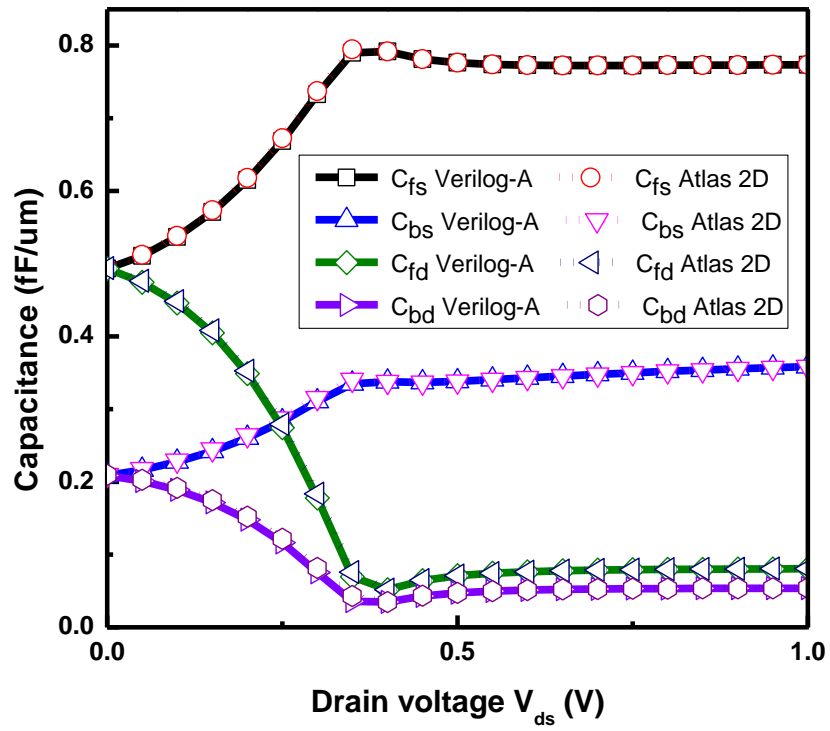


Figure 2.10. Capacitances vs. V_{ds} with $V_{bg} = -0.2$ V, $V_{fg} = 1$ V (Lower). L_g is 50 nm

2.3 Body-Biasing versus Independently Biased DGMOSFET

2.3.1 Reverse Body Biasing

Body biasing has been employed as a method for dynamically changing the threshold voltage of MOS devices. For NMOS devices, the V_{th} is increased when its body-source voltage is biased to be negative. This is referred to as reverse body biasing (RBB). In this technique, devices are fabricated for lower V_{th} than the design target and V_{th} is set to the target by adjusting the RBB. While RBB does allow for tuning of V_{th} , it also extends the drain-substrate depletion layer, which worsens the SCE. Furthermore, for deep submicron devices, the body effect coefficient γ is relatively small since the channel potential is more strongly influenced by the drain than by the substrate due to the DIBL effect. Coupled with SCE, the die-to-die V_{th} variation is increased by the substrate bias. This hampers adoption of RBB as an adaptive approach, especially at sub-50nm technology node.

2.3.2 Forward Body Biasing

On the other hand, the V_{th} is reduced when the body-source voltage is biased to be positive. This is referred to as forward body biasing (FBB). FBB is applied to a transistor with high V_{th} to bring V_{th} down to the target value. Since FBB alleviates the device short-channel effects, it reduces sensitivity of V_{th} to variation in gate length, oxide thickness, and channel doping.

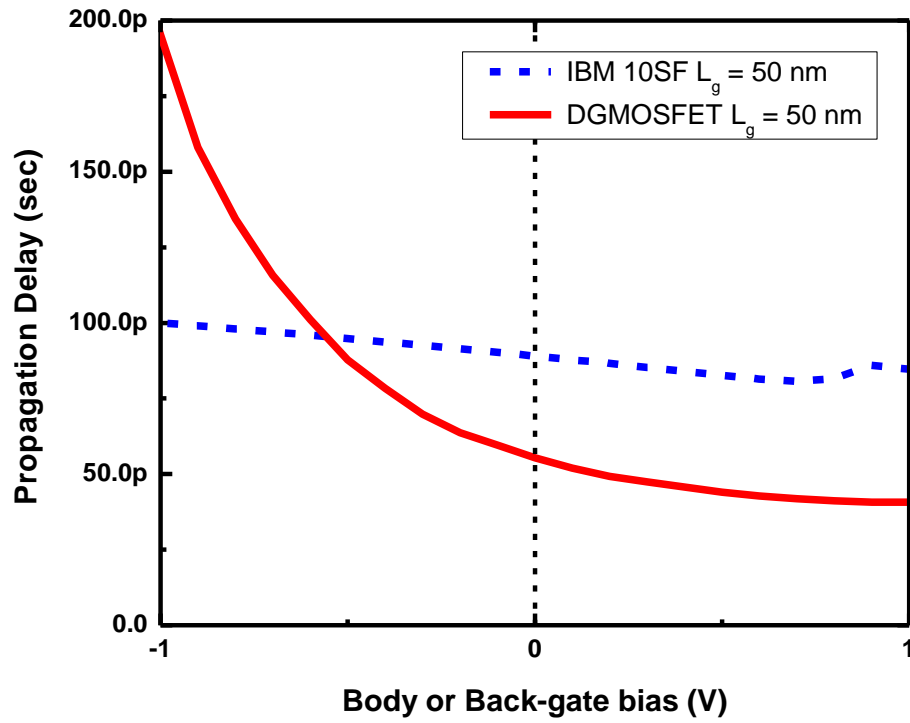


Figure 2.11 Propagation delay of 12 inverter chain vs. independently biased back-gate voltage and body-bias (solid line: DGMOSFET, dashed line: SGMOSFET with body-biasing).

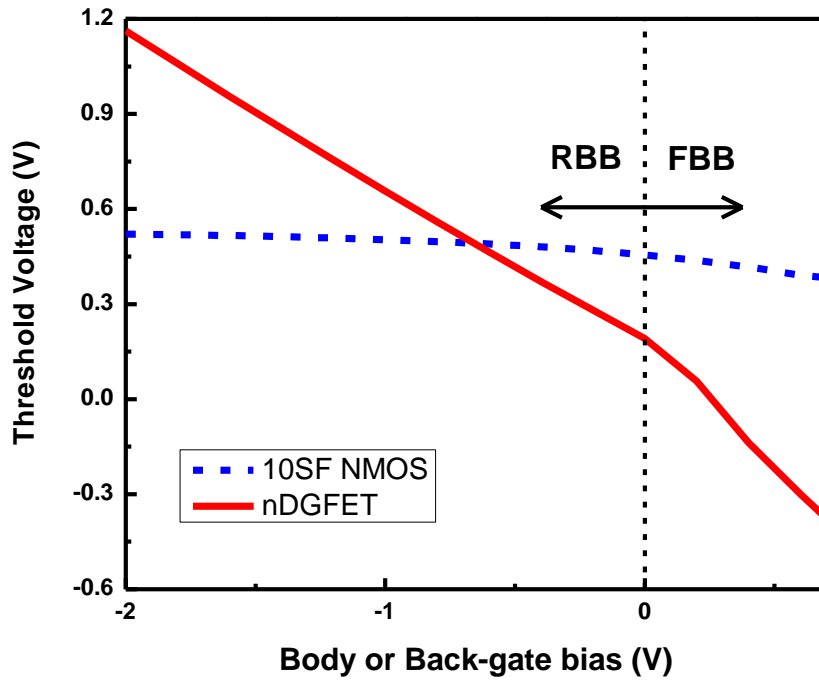


Figure 2.12 Comparison of threshold voltage tunability. Dashed line is a 50 nm nmos transistor of IBM's 10SF Technology and solid line is 50 nm DGMOSFET compact device model

Even though FBB improves circuit performance by lowering V_{th} , FBB increases leakage current due to parasitic bipolar current and forward source-body junction current. This limits the range of practical FBB values. Due to its robustness over SCE, FBB is reported to be effective down to 10 nm gate length MOSFETs [29]. However, the very narrow range of practical FBB values and problems such as forward biased pn junction current, parasitic bipolar transistor, CMOS latch-up phenomena, and the constraint of applicability to large blocks still limits its widespread use as an adaptation approach.

2.3.3 Independently biased Back-Gate

The threshold voltage of the DGMOSFET can be modulated by adjusting the independent bias on the back-gate electrode. If the back-gate voltage (V_{BG}) is reversely biased such that the back-channel surface is in accumulation, then this accumulation layer serves as an electrostatic screen for the effective leakage path in the OFF state and results in increase of the effective threshold voltage. Inversely, when the back-gate is forward-biased in weak inversion, the formation of the back-channel causes a decrease of the effective threshold voltage. Since our DGMOSFET is composed of undoped ultra thin Si body and has better confinement of electric field, this device does not have the problems that are present in the body biasing approach. Additionally, independently biased back-gate MOSFET provides a wider tuning range for V_{th} as shown in Figure 2.12. These superior characteristics of DGMOSFET provide the motivation for their use in the design of adaptive circuits in this work.

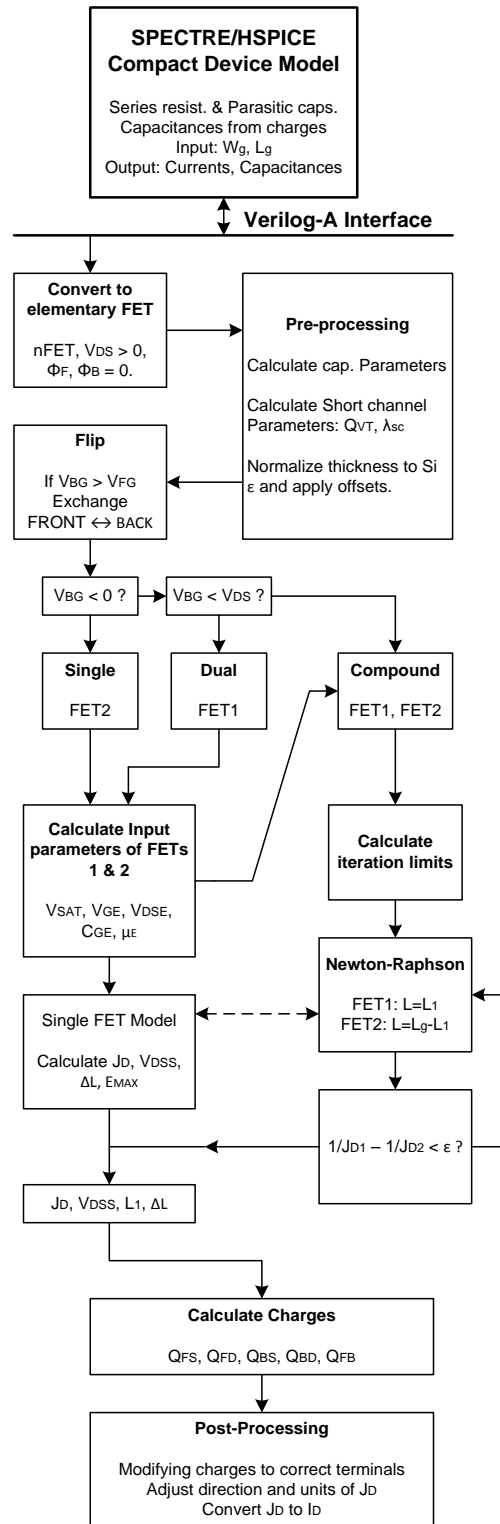


Figure 2.13 Modeling procedure for Verilog-A coding and Program flow control for mixed mode DGFET model.

2.4 Parametric Variation Compensation using Adaptive Circuits

This section discusses adaptive circuits implemented to compensate for variations in device parameters using independently biased DGMOSFETs. The propagation delay of critical path is observed to be reduced significantly with increasing back-gate bias of the DGMOSFET. This approach, suitable for nano-scale integrated circuits, where device variability increases, allows for the precise control of the overall VLSI circuit performance with a low overhead.

2.4.1 Design of Adaptive Circuit

The schematic illustrating the implementation of the proposed automatic adaptation circuit is presented in Figure 2.14. Generally, in adaptive control systems like DVS, critical path monitors have been used as part of closed loop adaptive circuits. Dedicated implementations of critical path monitoring have been reported in several applications [30, 31]. In this section, a simple critical path replica is inserted between the pipelined stage of the target circuits rather than a full-embedded implementation on the actual critical path. The automatic adaptation circuit consists of three blocks: (1) a Delay monitoring, (2) a Signal processing, and (3) a Back-bias generating block. The Delay monitoring block compares the timing delays of critical path and direct path. Initially, the ‘delay’ signal is generated from the main clock of the system. In this example, we have used a ‘delay’ signal with a frequency that is one eighth of the main clock frequency. That means the timing delay comparisons are performed at an

interval of 8 times the system clock. At each rising edge of the ‘monitor’ signal, this circuit detects the critical path delay of a DGMOSFET block where its back-gates are independently biased from the Back-gate generating block. When critical path delay is longer than the lowest allowed timing specification, the detector outputs to speed up the circuit. This case is explained in the timing diagram of Figure 2.17. On the other hand, when critical path delay is short enough to meet the timing requirement, the detector sends a signal to slow down the circuit, increasing its threshold voltage to maintain low-power consumption. The Signal processing Block increases and decreases its output number by accumulating the output digits (e.g. ‘+1’, ‘-1’, and ‘0’) from the Delay monitoring block. This output number can have a range between its lower and upper bound determined by the pre-defined programmable counter bit. Therefore, this range of output number defines a resolution of digital-to-analog converted feedback signal for generating back-gate bias. Finally, the Back-bias generating block translates the output number from the previous block to usable analog voltage for back-gate bias. The output from the block is directly applied to the back-gate bias of nDGMOSFET. The back-gate bias for the pDGMOSFETs inverts its polarity through the use of an inverting op-amp and level shifter that adds V_{dd} to the output. Using this feedback mechanism, the automatic adaptation circuit can automatically compensate for parametric variations and recover its performance. For fail-safe control and to prevent the oscillations at the converter output, a small delay (2 – 5%) can be added to the critical path [32]. This additional delay should be controlled to minimize the quantization error resulting from its counter bit.

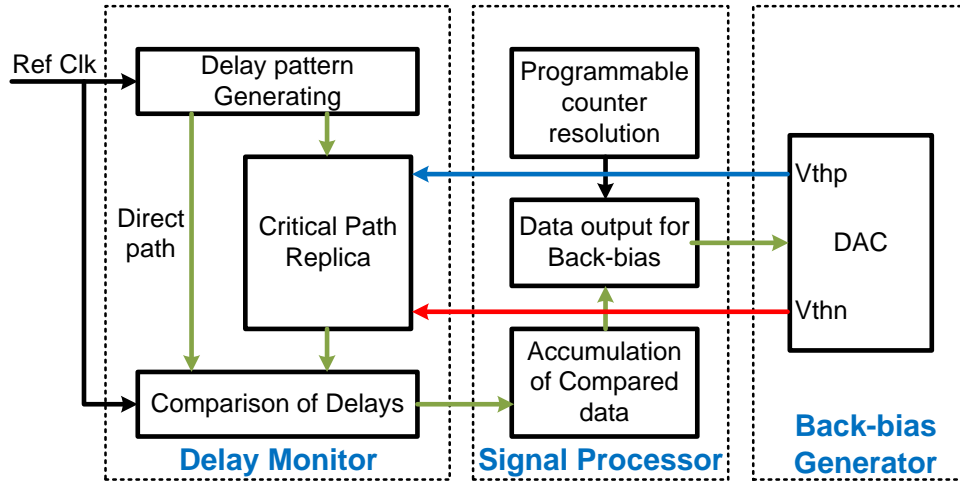


Figure 2.14 Scheme of Adaptive Circuit. Critical path replica is composed of nDGMOSFET and pDGMOSFET devices and mixed-signal control circuitry is implemented by IBM's 90 nm CMOS process technology.

(1) Delay monitoring Block

Figure 2.15 and 2.16 show the schematic of the Delay monitoring block. Because the direct path can always propagate the delay signal within a cycle of the system clock frequency even in the presence of worst case variations, it can be thought as reference timing specification for delay comparison. In the presence of variations, the signal at the output of the other paths may not be computed within a single cycle causing errors in the circuit. Therefore, comparing the signals at the output of each path allows us to determine whether the circuit meets the target timing specification.

(2) Signal processing Block

The Signal processing circuit is composed of an adder, a programmable counter, and simple logic for maintaining the output number within the upper and lower limits. The programmable counter determines an upper and lower bound and the adder

accumulates its internal number according to this limit and input from the Delay monitoring block. A detailed functional schematic is displayed in Figure 2.18 (a).

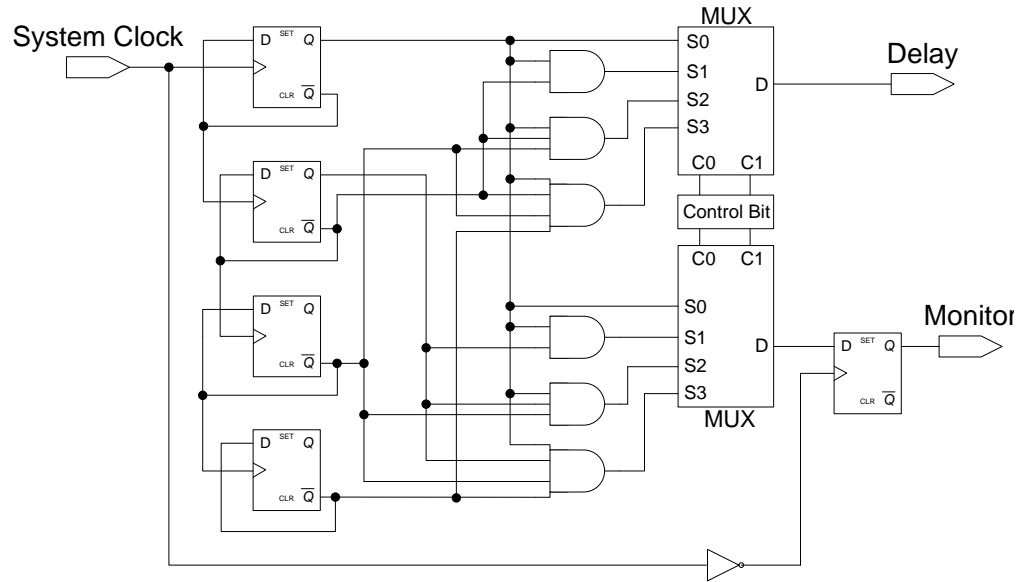


Figure 2.15 Schematic of Delay Pattern Generator with 2x, 4x, 8x, and 16x reference clock

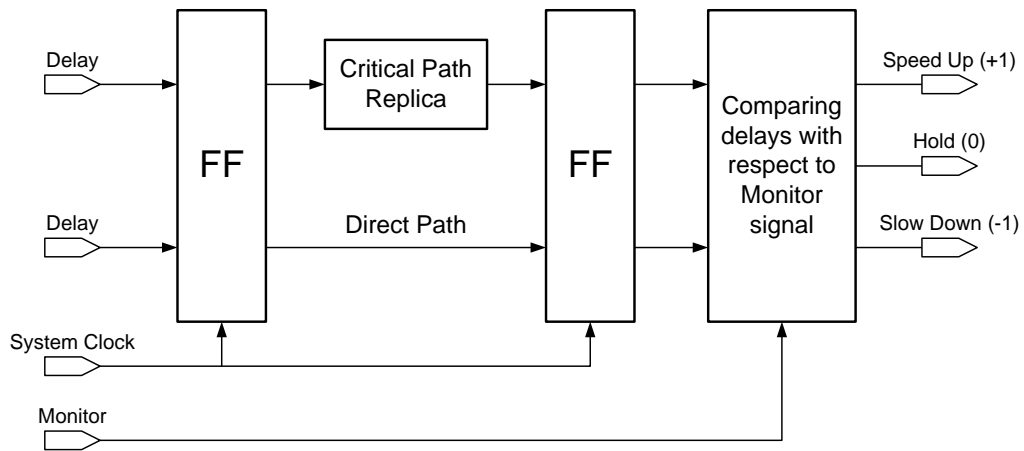


Figure 2.16 Schematic diagram of Delay Monitoring block. Delay pattern generator has flexibility between adaptation speed and dynamic power consumption of control circuitry by employing multiple frequencies of delay signal.

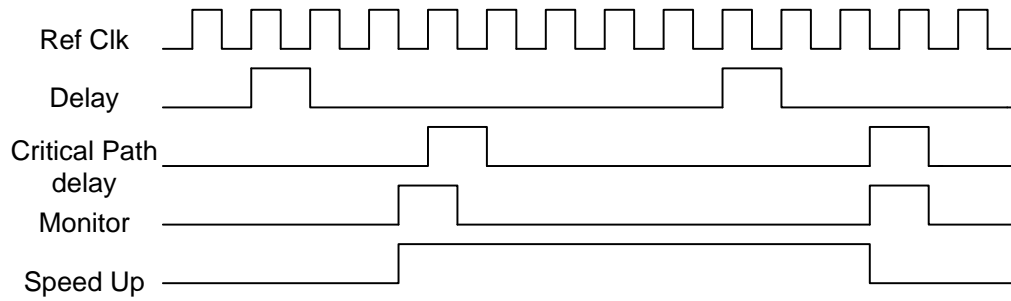


Figure 2.17 Error Detector Timing Chart in case of speed up. If the Delay signal passed through critical path replica is greater than required timing specification, Error detector outputs to speed up the overall circuit speed. Speed up signal is maintained until the timing requirement is met. In this timing chart, 8x Delay signal is used for error detection.

(3) Back-bias generating Block

The back-bias generating Block (Back-Gate Bias Generator) is divided into two parts; one is a typical digital to analog converter used to determine the back-gate bias of the nDGMOSFETs and the other part is a voltage translator which takes the output of the DAC and translates it to a suitable voltage level for the pDGMOSFETs. This is performed by using an inverting amplifier and shifting the DC level by adding a reference voltage, V_{dd} as shown in Figure 2.18 (b).

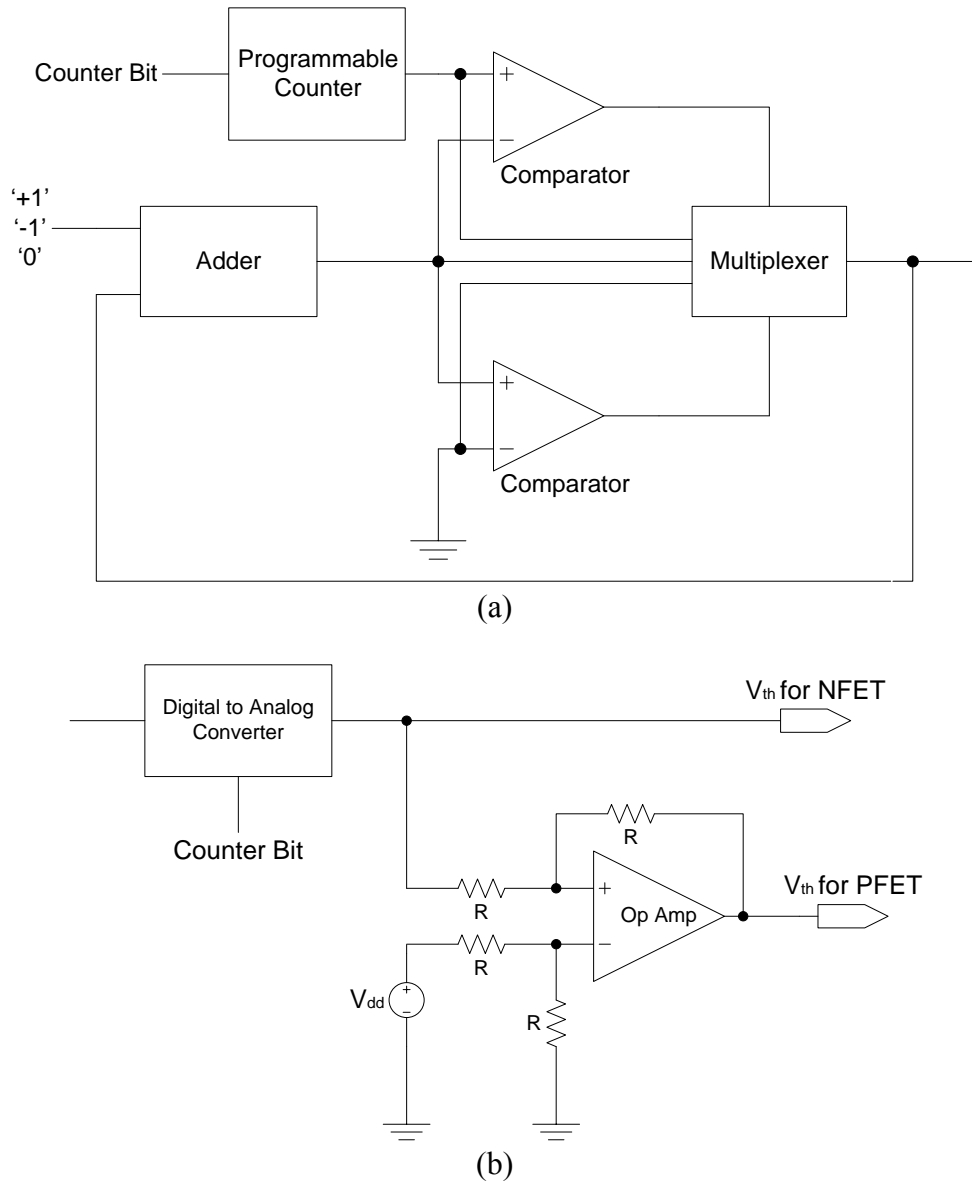


Figure 2.18 (a) Functional components for Signal Processor and (b) Back-bias generating Block.

2.4.2 Simulation Results

Simulation of the adaptive circuit was performed on the Cadence® Virtuoso platform with Spectre circuit simulator. IBM's 90 nm CMOS bulk device technology

was used for implementing other peripheral parts of the adaptive circuit. Figure 2.19 presents a simulation result that overall timing specification of the circuit is satisfied by reducing the critical path propagation delay, which reflect the total undesirable effects, including variations from process, temperature, and statistical contributions, as well as dynamic variations such as aging effects. The reference system clock frequency, f_{ref} , is chosen based on the given design specification, which means that the propagation delay of the critical path should be within the timing window of $1/f_{\text{ref}}$. For each generated test pattern, it is shown that the self-adaptive mechanism is activated to speed up or slow down the circuit. In this simulation, we see that the time required for the automatic adaptation circuit to fully adjust the threshold voltage is around 20 ns. If we increase the frequency of ‘delay’ signals, the time required for adaptive processing will be reduced. However, this will also result in increased power consumption. Since the threshold voltage tuning capability of DGMOSFET device are superior to body-bias method, the adaptation process will be accomplished in shorter times, which means the additional power overhead for adaptation can be maintained very small compared to the body-biasing.

While static compensation techniques such as clock tuning, VTCMOS (Variable Threshold voltage CMOS), and dynamic supply voltage can effectively compensate process variations, other variations such as temperature, voltage droops, noise, and transistor aging are dynamic and change throughout the lifetime of the circuit. These cannot be compensated using a static technique and are typically mitigated using either reduced frequency or higher supply voltage. This mitigation is expensive in terms of performance degradation and increased power consumption and is becoming

prohibitive as design margins shrinks. To achieve an energy-efficient microprocessor which operates correctly in the presence of these variations, a method of automatic threshold voltage adaptation by feedback control loop is necessary.

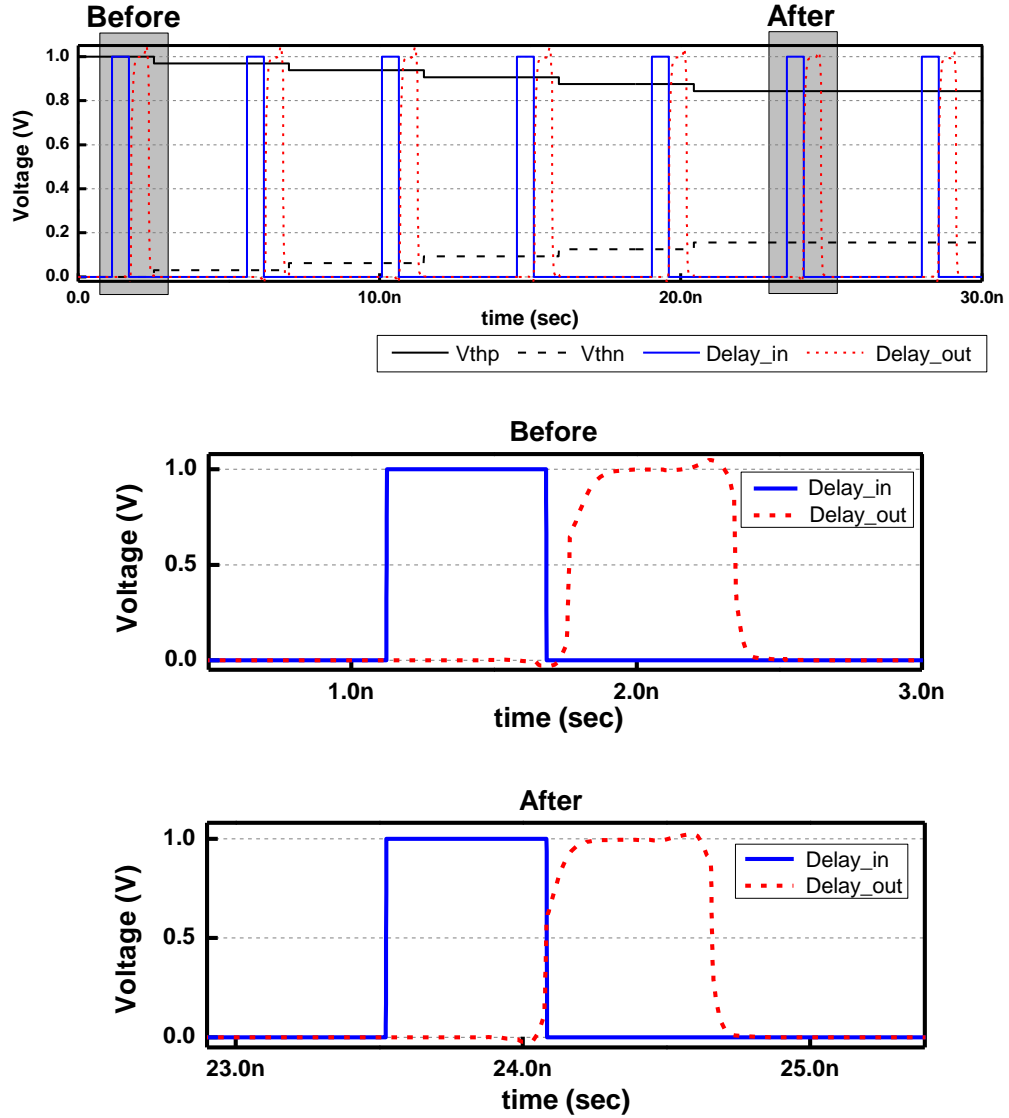


Figure 2.19 Critical path propagation delay is reduced to satisfy design specification by applying back-gate bias. The delay from critical path replica was initially 650 ps and reduced to 550 ps, which meets a timing specification imposed by f_{ref} , 1.8 GHz. Back-bias for nDGMOSFET and pDGMOSFET are changed from 0 V and 1 V to 0.84 V and 0.15 V, respectively. 20% of performance improvement was achieved by employing the above adaptive circuit design.

2.5 Power-Performance Adaptation

Since our adaptive approach using DGMOSFET has superior V_{th} tunability over conventional body-biasing scheme without the undesired side effects presented in RBB and FBB, a higher level of adaptation is possible using our design approach. This tunability, combined with the use of dynamic supply voltage scaling, allows us to significantly relax the hard trade-off between circuit performance and power consumption. Traditionally, voltage scaling for reduced power consumption has always been accompanied by performance degradation and this performance reduction has been partially compensated by decreasing V_{th} of the target devices. As shown in Figure 2.20, the performance degradation when the supply voltage is scaled to 0.8 V can be compensated only if the amount of threshold voltage reduction is greater than 130 mV. But this adaptation range is practically impossible with conventional body-biasing schemes. Figure 2.12 shows that the maximum possible decrease of V_{th} is just about 65 mV with RBB of 2 V or 50 mV with FBB of 0.5 V, which are practical limit of body-bias values for modern VLSI designs. However, if employing a DGMOSFET, this threshold voltage decrease can easily be achieved by applying an increase of 0.2 V back-gate bias as Figure 10 shows.

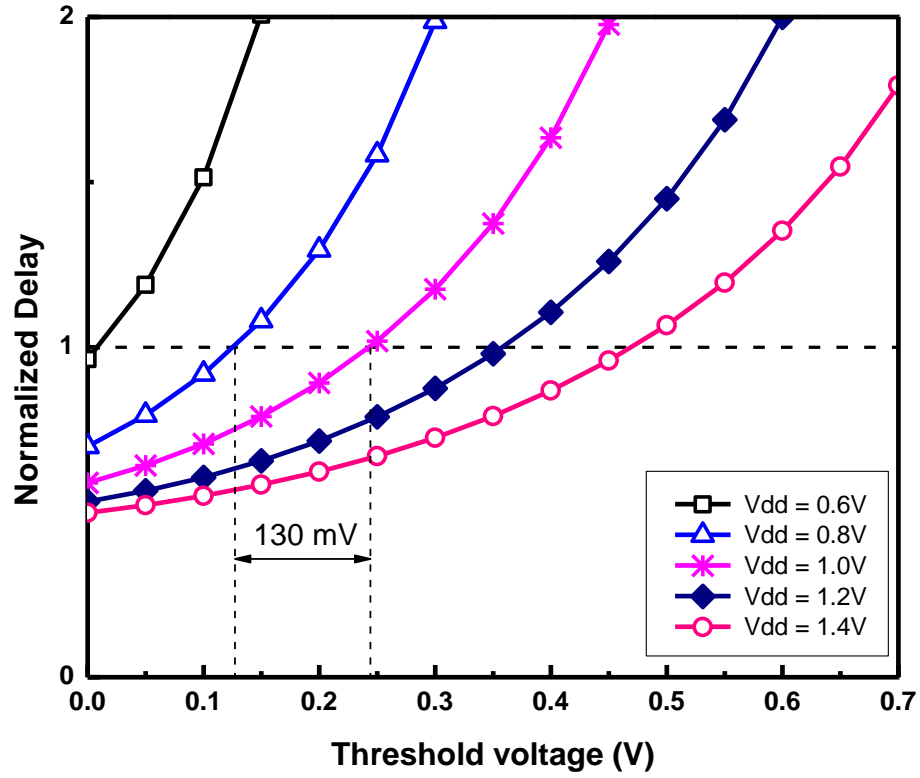


Figure 2.20 Normalized delay in terms of threshold voltage with different supply voltage values

The previous discussion shows how DGMOSFETs allow us to have a wider range of adaptive control of V_{th} . But the question then becomes, given a power budget and speed requirements, what are the optimal values of V_{dd} and V_{th} ? Figure 2.21 shows the simulation results of an 8-bit Radix-4 multiplier using 45 nm DGMOSFET devices. Through this plot, one can see the advantage of using dynamic V_{th} adaptation with DGMOSFETs. When adjusted for the same delay, the power consumption using dynamic V_{th} adaptation with a scaled V_{dd} of 0.8 V decreases by 40% compared to the case of V_{dd} only scaling. On the other hand, if we adjust for the same power consumption, dynamic V_{th} adaptation with a scaled V_{dd} of 0.9 V is 50% faster than what is obtained by using V_{dd} only scaling. For ultra-low power consumption or stand-

by status, a deeper V_{dd} scaling is observed to be required. Based on this observation, we can define four distinct operating regions depending on power budget and system performance. In region 1, referred to as “stand-by” or “sleep mode”, V_{dd} is scaled down to the lowest possible value and V_{th} scaled to the highest possible value, or the supply voltage can be turned off by power gating [33]. In region 2, called the “low power mode”, which is adequate for battery maximization of portable electronics, V_{dd} is reduced to the lowest value and then V_{th} is adjusted according to the given specification on propagation delay. In contrast to the other regions, the lowest scaled V_{dd} in this operating region is critical to reduce power consumption. In region 3, the so-called ‘balanced-mode’, to have an optimization of speed and power, V_{dd} is slightly reduced from the nominal value and then V_{th} adaptation finds the operating point that provides the required performance. Finally in region 4, or the ‘high-speed mode’, V_{dd} is set to the nominal value and V_{th} is adaptively reduced to meet the given timing specification. Based on this observation, we summarize the four operating modes as described in Figure 2.22. We denoted each operating mode by a two-bit binary number. The transition between each node can be activated by user input or by demand of operating system. According to this mode information, timing specifications are determined and the reference frequencies are generated from the Digital PLL.

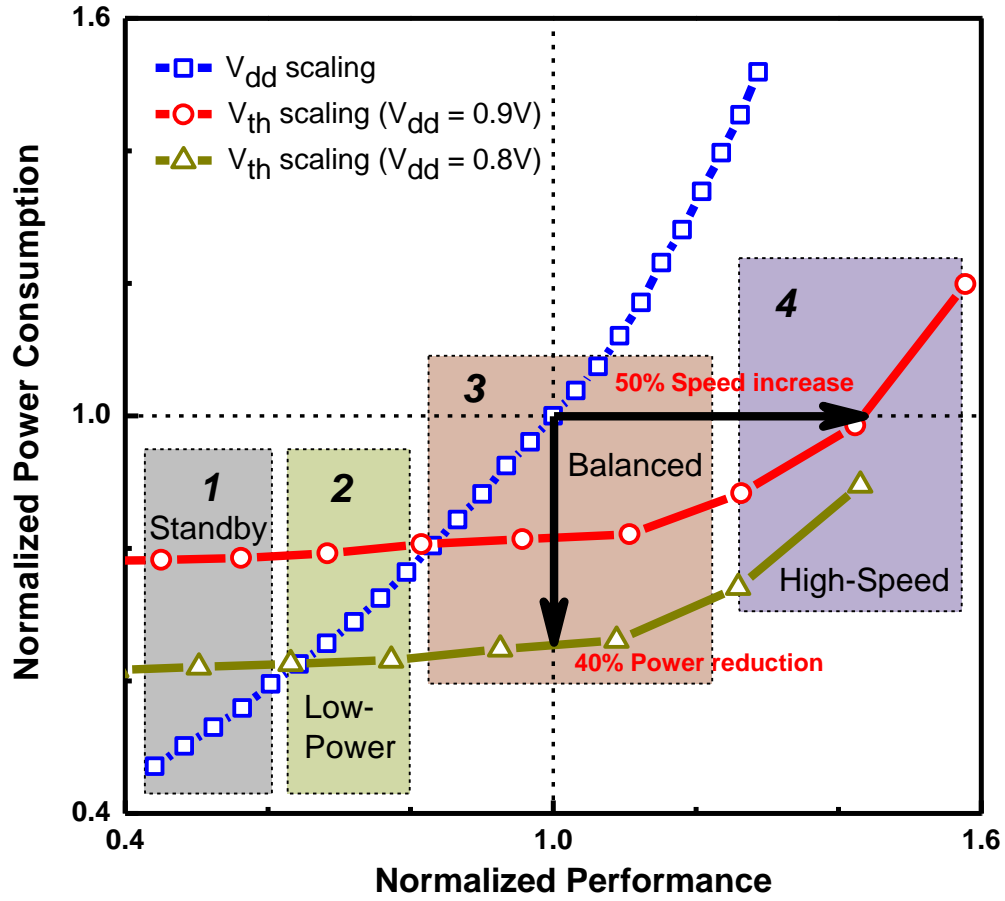


Figure 2.21 Comparison of V_{th} adaptation and V_{dd} scaling. Power consumption and delay are calculated from an 8-bit Radix-4 multiplier designed with DGMOSFET devices of $L_g = 40$ nm.

There are multiple ways to implement the above scheme to reduce power consumption while maintaining the required operating frequency: (1) employ pre-defined combinations of multiple power supplies to lower V_{dd} and multiple threshold voltage to reduce leakage current: sets of V_{dd} values and V_{th} values are appropriately chosen by selected operating modes, (2) mix pre-defined sets of multiple supply voltages and dynamic adaptation of DGMOSFET as used for compensating variations

in section IV, and (3) employ fully automatic, two independent adaptation loops for both V_{dd} and V_{th} . Most of recent technologies provided by major semiconductor foundries support multiple threshold voltage devices and different supply voltage domains in a single chip. And some research groups have suggested an implementation of dynamic adaptation by using body-biasing [34-36].

In this section, we propose another higher level of adaptive circuit design to implement an optimized adaptation between energy and performance. A schematic of this circuit is shown in Figure 2.23. This circuit is composed of two independent adaptation loops for V_{dd} and V_{th} , respectively, a critical path delay comparison block, Leakage Current Monitor (LCM) circuitry, and other peripheral circuits for generating control signals and various clock frequencies. The approach is similar to the circuit presented in section IV. The difference is the addition of V_{dd} adaptation loop, LCM circuitry, and control signals for each operating mode. Let us assume circuit operation starting from a standby or sleep mode. In this mode, there is no need for dynamic adaptation, so we just set V_{dd} and V_{th} to their lowest and highest value, respectively, or we activate power-gating to turn off the entire block. According to the mode information status, V_{dd} and V_{th} generators in Figure 2.23 output the pre-defined lowest and highest values to the target circuit or generate control signal to turn off supply voltage. In the high-speed mode, based on our observation from Figure 2.21, we activate only the threshold voltage adaptation loop with nominal supply voltage value (in this example, $V_{dd} = 0.9$ or 1 V) since dynamic threshold voltage adaptation shows superior control over supply voltage scaling especially in the high-speed operating region.

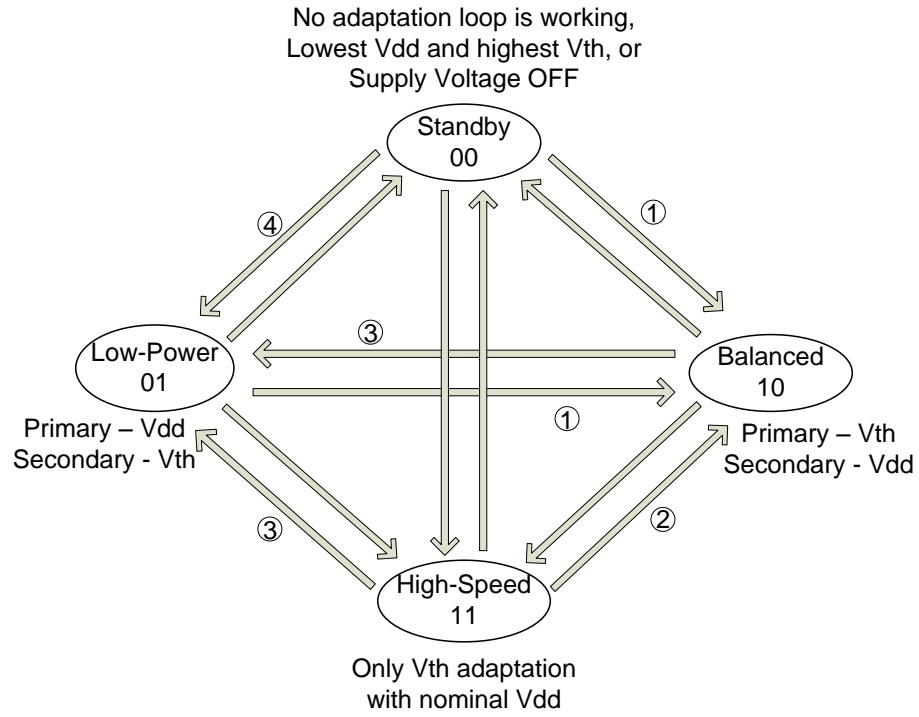


Figure 2.22 Summary of operating modes for optimized V_{dd} and V_{th}

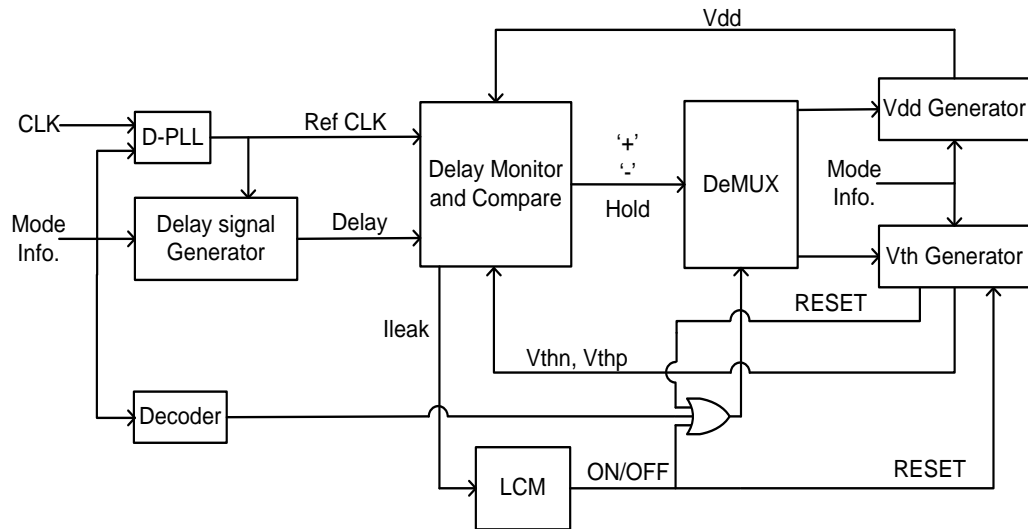


Figure 2.23 Schematic diagram of circuit design for V_{dd}/V_{th} adaptation.

Up to this point, our implementation is straightforward and utilizes the approach highlighted in the previous section. Next we discuss the implementation of more complex algorithms for the other two operating modes. For this purpose, two independent control loops must be employed simultaneously. How can we combine these two control loops to find the optimized values of V_{dd}/V_{th} for “Balanced mode” and “Low-power mode”? The answer to this question also comes from the simulation results shown in Figure 2.21. In operating region 3 (Balanced mode in Figure 2.22), threshold voltage control should come first so that power dissipation is minimized while maintaining circuit performance. Therefore, in this operating region, we employ a V_{th} adaptation loop as the priority control loop for finding the optimized operating point. And then, we employ supply voltage adaptation as the second control loop. However, we must have a lower limit for V_{th} to prevent leakage power from dominating total power dissipation. The subthreshold leakage current can be monitored by an LCM [37]. LCM generates ON/OFF signal to switch the priority of control loops and sends a reset signal to V_{th} generator to initialize threshold voltage adaption loop. This process is explained in detail in Figure 2.24 (1, 2). At time A, timing delay still doesn’t meet the required specification even with decreasing its threshold voltage to the lowest possible value. In this case, V_{th} generator undergoes self-reset and initializes its loop and control priority is transferred to V_{dd} adaptation loop at the same time. Since V_{dd} adaptation loop has the control priority and V_{th} value reset to the highest value, V_{dd} increases to meet the timing specification. And then control priority goes back to the V_{th} loop because RESET signal from the V_{th}

generator is held for very short time. At time B, LCM outputs an ON signal since leakage current exceeds the maximum allowable current value, so resets and initializes the V_{th} generator. Since V_{th} value returned to the highest value, LCM outputs OFF again and sends the control priority back to the threshold voltage adaptation after a small increase of supply voltage. Finally at time C, this adaptation circuit finds the optimized operating point and stays at the determined V_{dd} and V_{th} value. Similarly, in case of timing surplus, V_{th} is increased to the highest possible value until finding an operating point to meet the timing specification. At time D, if there is still timing margin, V_{th} generator is initialized and V_{dd} takes the control priority. And then adaptation circuit completes its adaptation at time E.

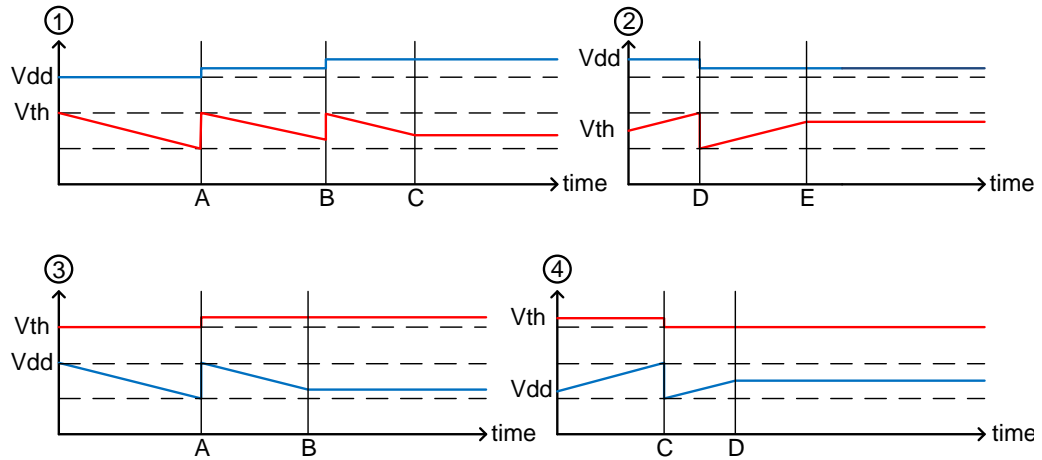


Figure 2.24 Adaptation of V_{dd} and V_{th} in operating region 3 and 2, respectively

For operating region 2 in Figure 2.21 (Low-power mode in Figure 2.22), the lowest supply voltage is preferable to guarantee the maximum battery life. Therefore, in contrast to region 3, supply voltage adaptation must have a priority control over threshold voltage. With this concept in mind, we implement similar mechanism for

optimized V_{dd} / V_{th} value. This is also shown in Figure 2.24 (3, 4). Finally, we design the peripheral circuitry required to make an automatic transition between each operating mode depending on the two-bit mode information input, shown in Figure 2.23. The D-PLL is digital phase-locked loop, which generates various reference frequencies to be compared with critical path delay. The core of the Delay monitor and compare block is same as the Delay monitoring Block presented in section IV. We add a functionality to select a divider number according to the mode information. For example, when there are two independent adaptation loops, we divide reference clock by 4 instead of 8 to speed up the time required finding the optimized V_{dd}/V_{th} values. A decoder is used to initially determine the priority of adaptation loops from the mode information input. Transitions between each operating mode are denoted as circled numbers and Mode information bits are also designated as shown in Figure 2.22. These numbers show the adaptation mechanism as depicted in Figure 2.24.

2.6 Adaptive Circuit Design for Improved Noise-Margin

To ensure circuit robustness, even with the increased variability that results from device scaling, the noise margin is held relatively constant between successive technology generations. This is the primary reason why the voltage levels scale slower than transistor dimensions. This in turn limits the reduction in power consumption that can be obtained by moving to more advanced technologies. Evident here is a relatively strict trade-off between noise margin and power consumption. Dynamic voltage scaling does little to relax this trade-off. In fact, even when supply voltages remain

constant between technology nodes, dynamic voltage scaling or supply voltage adaptation results in significant reduction of noise margin in digital designs. We have explored methods of alleviating this problem by using independently biased back-gated DGMOSFETs as circuit elements. Comparisons are drawn between these designs and identical designs with equivalent single-gate silicon-on-insulator devices to understand the impact on overall noise-margin, power consumption, and adaptation. We also suggest an adaptive circuit design to maximize noise margin at the expense of slight increase in design complexity.

In digital design, a measure of the sensitivity of a gate to noise is given by the noise margins NM_L (noise margin low) and NM_H (noise margin high), which quantize the size of the legal “0” and “1,” respectively, and set a fixed maximum threshold on the noise value [38]:

$$NM_L = V_{IL} - V_{OL}, \text{ and}$$

$$NM_H = V_{OH} - V_{IH}.$$

A typical inverter voltage-transfer characteristic (VTC) looks like a butterfly shape and the noise margin can be represented as the area of this curve [39]. Lowering V_{dd} to save power dissipation reduces the signal swing and makes the design more sensitive to external noise sources or thermal noise that do not scale. Previous work has shown that even in the case of ultra-low power budget, supply voltages less than 0.6 V are not recommended due to noise margin degradation [40]. The question then becomes whether there is a way to decrease power consumption while maintaining a large noise margin. One such method is threshold voltage adaptation. As seen in Figure 2.24, increasing the threshold voltage leads to improved noise margin as well as reduced

power consumption. This comes at the cost of performance degradation, but adaptation of threshold voltage is the only way to reduce power consumption while maintaining noise margin

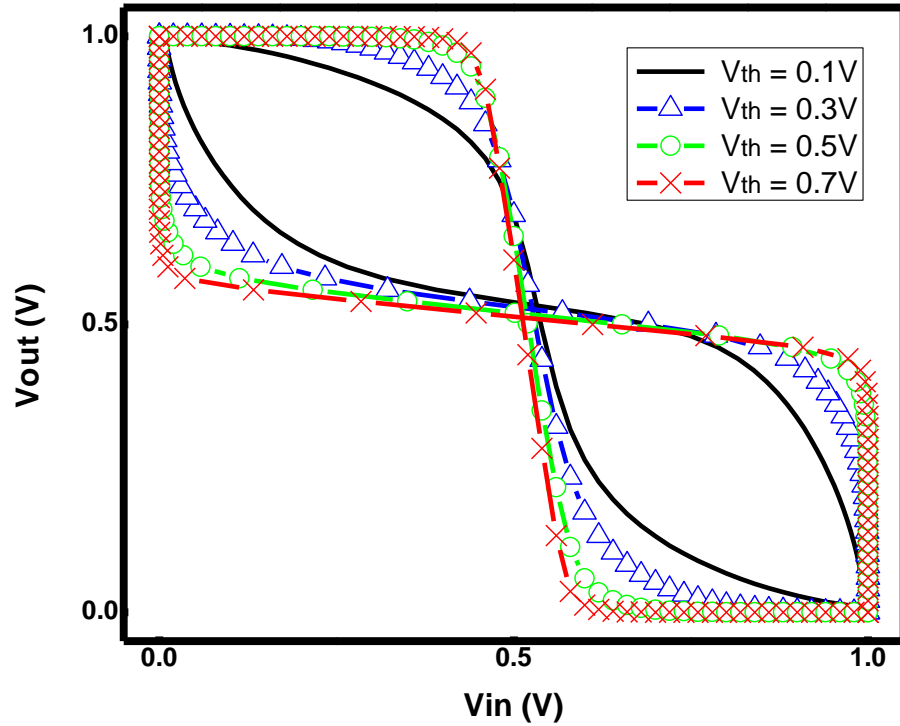


Figure 2.25 Noise margin of a unit-sized inverter. IBM's 45nm SOI process technology is employed to illustrate noise margins with varying threshold voltages. Note that noise margin increases as threshold voltages increases.

To demonstrate the effectiveness of noise-margin adaptation using DGMOSFETs, we designed two 16-bit, Radix-4, tree-multipliers with 4:2 compressors, one using 45 nm DGMOSFETs and the other using 45 nm SOI SGMOSFET devices. We compare the critical path delay and power consumption of each multiplier block while varying the supply voltages of the SGMOSFET-based multiplier and the back-gate biases of the DGMOSFET-based multiplier. The simulation results are plotted together in

Figure 2.26 with the noise margins of each multiplier block. Three distinct operating regions can be considered for comparing the DGMOSFET-based and SGMOSFET-based multipliers in terms of performance, power, and noise margin. In the upper-left region, DGMOSFET-based design may be appropriate for ultra-low power budgets while maintaining circuit noise-margin, e.g., sensor networks located in extreme noise condition. In the lower-left region, the back-gate of DGMOSFET is biased negatively to reduce its leakage power. The back-gate is as strong as the front-gate, and therefore the device has degraded sub-threshold slope and transconductance due to the capacitive division of the channel potential between the two gates. As a consequence, the critical path delay of the DGMOSFET multiplier is slower in this entire region. Simultaneously though, the noise margin when using DGMOSFETs in this region is much larger than when SGMOSFETs are used. These results indicate that depending on the power constraint, timing specification, and error tolerance of the required application, one technology may be preferable over the other. Finally, when high power consumption is allowed, both technologies are comparable, but the SOI SGMOSFET devices are preferable since implementation is cheaper and simpler.

In the simulations of Figure 2.26, note that only V_{th} tuning is employed for optimization of energy, performance, and noise margin. As stated earlier, a reverse-biased back-gate results in a reduction of carrier mobility and thus degrades transconductance. Therefore, it may not possible to simultaneously optimize these three parameters in most cases. However, these simulations show that the adaptive tuning of V_{th} in DGMOSFETs allows them to span a wider practical range in energy-performance space over SGMOSFETs. Thus, adaptive tuning methods with

DGMOSFETs would be ideal candidates for combination with supply voltage scaling and may alleviate the tradeoffs between these three key metrics for a wide range of applications.

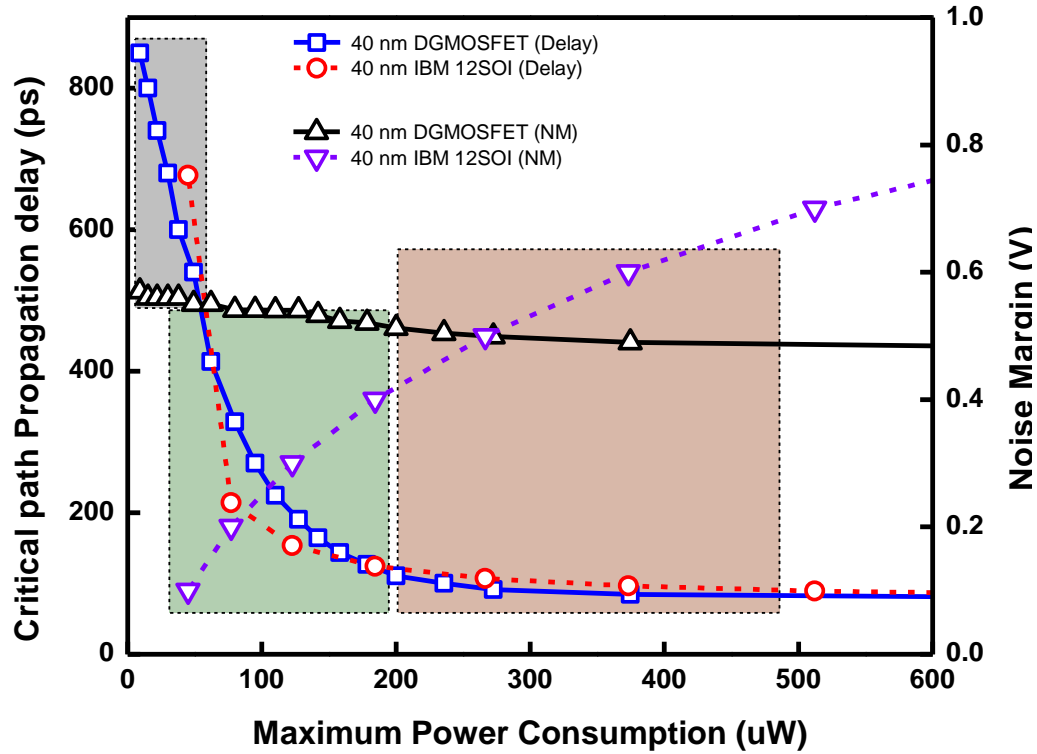


Figure 2.26 Power vs. NM and Power vs. Critical path delay of multiplier blocks. Each 16-bit multiplier is implemented with 45 nm DGMOSFETs and IBM's 45 nm SOI devices, respectively. Note that the noise margin of DGMOSFET is maintained since the power reduction is achieved by applying back-bias rather than scaling supply voltage.

Generally in digital design, the effect of changing the W_p -to- W_n ratio is to horizontally shift the transient region of the VTC. Increasing the width of the PMOS or the NMOS moves a switching threshold voltage V_M toward V_{dd} or GND, respectively. This property can be very useful, as asymmetrical transfer characteristics are actually desirable in some designs. For example, an inverter designed with a higher

switching threshold voltage can suppress the noise from the incoming signal which has a very noisy zero value. However, changing the switching threshold by a considerable amount using this approach is not easy. In the IBM's 45 nm SOI technology with supply voltage of 1 V, moving the switching threshold by 225 mV requires a transistor ratio of 100, which is prohibitively expensive. This is shown in Figure 2.27, which is plotted on a semi-log scale. Note that the nominal switching threshold is around $V_{dd}/2$, in this case 0.5 V.

In contrast, shifting the switching threshold voltage is easily accomplished by using independently biased back-gated DGMOSFETs, as displayed in Figure 2.27. By simply applying different back-gate bias voltages, we obtain an inverter with asymmetric transfer characteristics without increasing the circuit area. In addition, this method allows for post-fabrication adjustment of the transfer characteristics, something that cannot be achieved by adjusting the width ratio. We can make the transfer characteristics adaptive for the design specification or external noise, a property useful in several circumstances. Using these advantages of the DGMOSFET listed above, we suggest a buffer or inverter circuit to effectively filter the noise present in incoming signals. Details of the inverter circuit are shown in Figure 2.28. Depending on the voltage level of the input signal, the back-gate biases of the DGMOSFETs are multiplexed between two values, e.g. 0 V and -0.8 V. At the input level of supply voltage (1 V), nDGMOSFET is biased with a back-gate voltage of 0 V, which pushes the switching threshold down to 0.33 V so that the noise margin becomes 0.67 V. When the input level is at GND (0 V), the back-gate of the pDGMOSFET is biased at -0.8 V, changing the noise margin to 0.83 V. Therefore,

this inverter can reject up to 0.67 V of supply voltage noise and 0.83 V of GND noise. A simulation result showing the function of noise rejecting is exemplified as in Figure 2.29.

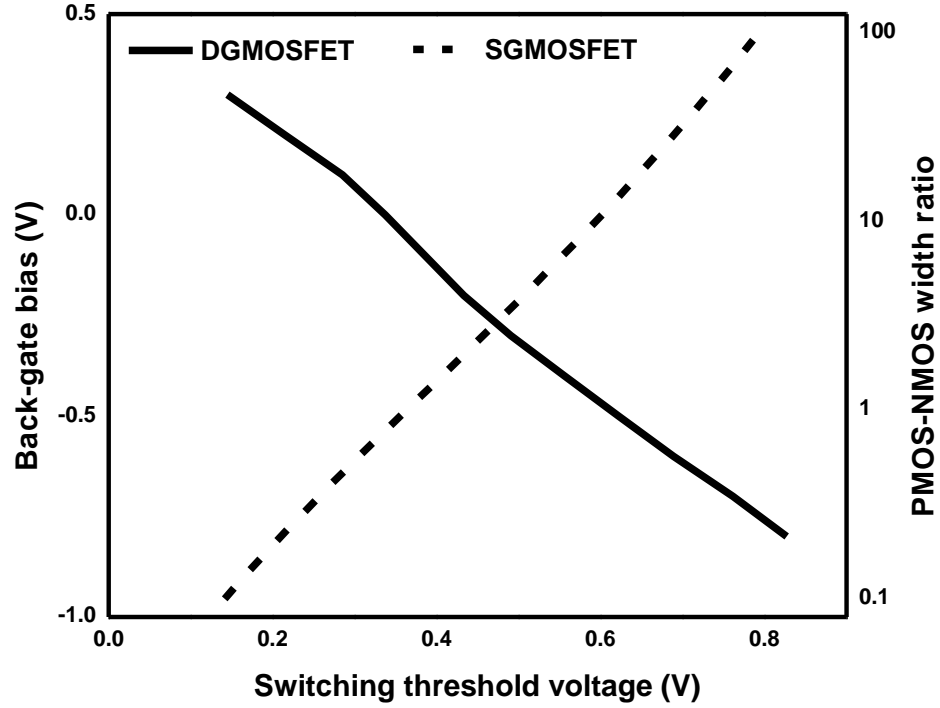


Figure 2.27 Simulated inverter switching threshold voltage versus PMOS-to NMOS width ratio (45nm SOI, $V_{dd} = 1V$, dashed line). Simulated inverter switching threshold voltage versus back-gate bias voltage (45nm DGMOSFET, $V_{dd} = 1V$, solid line).

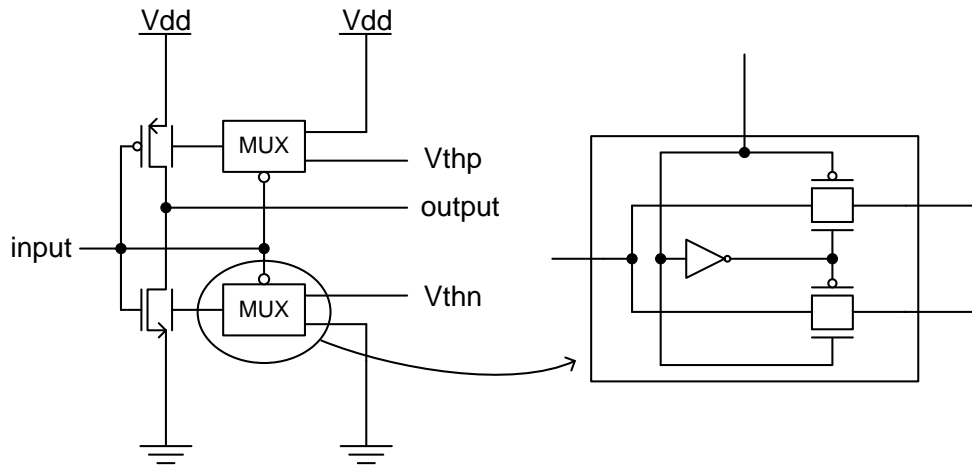


Figure 2.28 Inverter circuit for noise filtering, designed using 45nm DGMOSFETs and SOI SGMOSFETs.

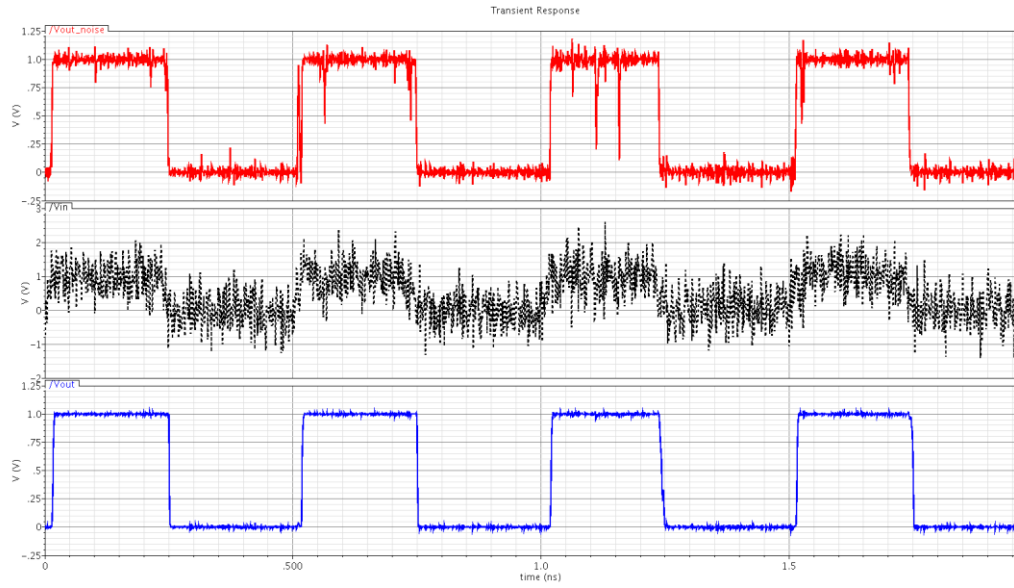


Figure 2.29 Output signal comparisons (Top: Standard buffer output, Middle: Input signal with noise, and Bottom: DGMOSFET buffer output).

2.7 Chapter Summary

Adaptive circuits using DGMOSFETs present a number of advantages in a variety of attributes in the sub-50 nm device and process technologies.

Some of these are:

1. *Compensating PVT variations*: We've shown that our adaptive design can compensate the parametric variations effectively.
2. *Solutions to chip aging and wear out*: Performance degradation resulting from this long-term variation can be compensated and recovered to original performance.
3. *Robustness to SCE*: DGMOSFET devices provide improved robustness against

Short-Channel Effect.

4. *Power-Efficient design methodology*: Adaptation between power and performance is more flexible with design choices and ultra-low power capability.
5. *Improvement of noise immunity*: It is shown that noise immunity is improved without sacrificing substantial performance degradation.

We proposed a new adaptive circuit design approach using DGMOSFET devices. This enables the user to control the threshold voltage of the target transistors after fabrication by applying a separate bias to the back-gate of the devices. This tuning of threshold voltages has two main advantages over conventional body-biasing schemes. The first is a significant improvement in the threshold voltage tuning range (11x that of conventional body-biasing schemes), and the other is that unwanted side effects that arise as a result of body bias adaptation are alleviated in the proposed adaptation scheme. Using this adaptive circuit design approach, we have demonstrated the capability of compensating both static and dynamic variations. This is possible by monitoring critical path delay which can be thought of as a metric that takes into account all relevant variations, and adjusting accordingly the threshold voltage to speed up or slow down the circuit to meet the given timing specifications. We also present a possible system-level design demonstrating how this threshold tuning scheme using DGMOSFETs can be combined with adaptive supply voltage scaling to allow the user to simultaneously optimize for both power consumption and performance according to the application-specific workload and power requirements.

Our simulation results using a 45 nm CMOS technology indicate that this adaptive circuit design can provide 50 % higher performance for the same energy, or 40 % less energy for the same performance. Increasing the threshold voltage leads to improved noise margin as well as reduced power consumption. This, of course, comes at the cost of performance degradation, but adaptation of threshold voltage is the only viable way to reduce power consumption while maintaining noise margin. An inverter design was presented as one possible application using this feature which cuts off the noise signal that peak up to 70 % of the supply voltage level. The proposed adaptation strategies using DGMOSFETs would allow designers to produce variation-free, high noise-margin circuits with workload awareness that allows for flexible tradeoff between power consumption and performance at sub-50 nm technology nodes.

CHAPTER 3

INEXACT COMPUTING USING PROBABILISTIC CIRCUITS

3.1 Motivation and Background

With highly integrated circuit designs at sub-50-nm process technology nodes, reliability issues resulting from PVT (process, voltage, and temperature) variations, aging effects, soft errors, and noise are major impediments to leveraging the benefits of device scaling. These problems have traditionally been addressed by incorporating a safety margin through operation at conservative voltages. As a consequence, the supply voltage in modern process technologies is significantly higher than originally suggested by constant field scaling theory [8]. This in turn, coupled to leakage and static power, has led issues of high power density. The increasing leakage currents puts a lower bound on the threshold voltages, and this in turn severely impedes further scaling of the supply voltages. As device dimensions have shrunk to sub-50-nm, increasing variations have widened the required margin for reliable operation. Due to the increased safety margin, performance enhancements gained from further device scaling are not fully exploited. Recent work proposing adaptive circuit design techniques [12], [41] allows designers to partially relax these safety margins by dynamically adjusting system parameters such as supply voltage, body bias, and operating frequency, and relaxes the strong trade-off between performance and power for optimized energy consumption. These methods cannot fully eliminate such margins since they must guarantee computational correctness in all cases including the worst-case combinations of extreme variations and inputs. As a further means to relax

these margins, methods employing in situ error detection and correction have been proposed by researchers [42], [43]. While some of the potential benefits of error detection and correction circuits have been highlighted in previous work [44], [45], their implementation requires a significant amount of additional clock energy and the error correction circuits are susceptible to metastability and have a substantial design overhead.

There are numerous applications where a low level of error can be tolerated. For example, in human vision, a certain level of information is sufficient to saturate the human visual system [46]. This is also true for human hearing whose good example is acceptance of wireless phones. Decision making system based on statistical models, Bayesian, and compression applications are common examples where error tolerance is intrinsic to the computation approach. This paper specifically addresses the power-error trade-offs that can be achieved in such computational tasks using mainstream logic elements. So, the argument is that power savings can be achieved and the extra cost to detect and correct errors or inserting safety margins to guarantee correct functionality can be avoided for many applications through new hardware accelerators.

A statistical approach is employed here to characterize a variation-aware nanometer CMOS digital logic using error probability and power consumption as the main metric. Since this approach does not require additional overhead, its use has the potential to reduce cost per die and critical path delay. Furthermore, relaxed safety margins allow more room for designers to aggressively scale power consumption by lowering the

supply voltage or to increase circuit performance by clocking at higher operating frequency. Even though this methodology may not be adequate for applications such as those where exactness is essential, there still are a large class of applications that use statistical performance metrics which would benefit from this methodology. Examples in our daily life of these include the signal-to-noise ratio (SNR) in audio and video signal manipulation, bit-error-rate in digital data communications, search engines in which results are displayed through a statistical metric, and others where errors, so long as they are within certain constraints, are acceptable. Statistical data processing and computing, also referred as “*Inexact Computing*”, is gaining popularity as an alternative approach for addressing reliability and power issues [47], [48]. By viewing nanoscale circuits as noisy communication channels, communication-inspired design techniques based on statistical estimation and detection algorithms are proposed and demonstrated [49]. In the approach proposed by Palem [50], a framework for probabilistic switches and computational models based on these switches that treat noise as a circuit design parameter were outlined. These models are employed here to show that probabilistic algorithms hold potential for low-energy computation.

In this chapter, we further explore and develop the details of a probabilistic approach in the context of device modeling and circuit design. A new circuit-level characterization and simulation methodology is demonstrated using EDA tools with IBM’s 45 nm 12SOI process technology. Specifically, the relationship between error probability and different types of noise coupling with varying noise RMS (Root Mean Square) values and circuit topologies is analyzed. Starting with a probabilistic CMOS

inverter as the basic building block, the methodology is applied to the gate-level implementations of systems with increasing complexity, culminating in a specialized 32-bit adder circuit which is a core building block in high-end processors for handling media-intensive data. The rest of the paper is organized as follows. Section 3.2 is a description of the probabilistic approach and Section 3.3 presents an experimental framework for validation. In Section 3.4, the dominant impact of input noise error probability in different situations is discussed. Section 3.5 investigates the relationship between errors and power consumption (or energy) by adjusting supply voltages and threshold voltages of the target devices. In Section 3.6, a novel MSB-LSB weighted, ultra low-power 32-bit CCS-CSS (Conditional Carry Select – Conditional Carry Sum) adder design is explored to show an effective way to exploiting the benefits of inexact computing for energy-efficiency in the presence of an extremely high-degree of unreliability. We conclude by summarizing the findings in Section 3.7.

3.2 Probabilistic Approach for Non-deterministic CMOS Logic

Statistical performance metric describing a characteristic of non-deterministic CMOS circuit due to the variations is presented in this section. A new framework using probabilistic approach which was originally suggested by Palem [50] is used and extended to explore the benefits of the *Inexact Computing*. This approach may seem to resemble to Goguen’s logic of inexact concepts [51], since a probabilistic distribution might be thought as representing multi-valued logic. But, the manipulations allowed in probability theory are different from those examples for fuzzy sets such as

manipulations of vagueness, ambiguity, and ambivalence rather than likelihood. In our probabilistic framework, noise source is employed as a model of variations in CMOS device technology. And, random Gaussian noise distribution is embedded to evaluate the properties of probabilistic circuit design.

3.2.1 CMOS Logic Implementation using Probabilistic Approach

As the simplest form of CMOS logic circuits, a non-deterministic inverter output can be described statistically. Probabilistically, the logic output is erroneous with a probability p and correct with a probability of $1-p$. If the binary values of the output and input of the inverter are denoted by out and in , respectively, then

$$out(t_f) = \begin{cases} \overline{in(t_i)}, & \text{with probability } 1-p \quad (\text{correct}) \\ in(t_i), & \text{with probability } p \quad (\text{erroneous}) \end{cases} \quad (1)$$

To illustrate how this is different from the traditional deterministic approach, the output waveforms of a deterministic CMOS inverter and an output with an input noise for the same input waveform are shown in Fig. 3.1. Since the probability of erroneous operation, p , is assumed to be a result of mixing all variations that adversely affect the output, the probability p is varied by changing the characteristics of noise source, the supply voltage, or the device parameters of the inverter. As a result of this unreliability, represented through introduction of noise, incorrect switching occurs at the output of the inverter as shown in Fig. 3.1. In modeling the noise, the approach of Stein [52] is employed where noise is assumed to be a random process characterized

by an Additive White Gaussian (AWG) distribution with zero mean and a standard deviation of σ . For this work, the noise is considered to have an Additive White Gaussian Noise (AWGN) characteristic over the entire frequency range of interest. 500GHz is chosen to be adequate as the upper bound.

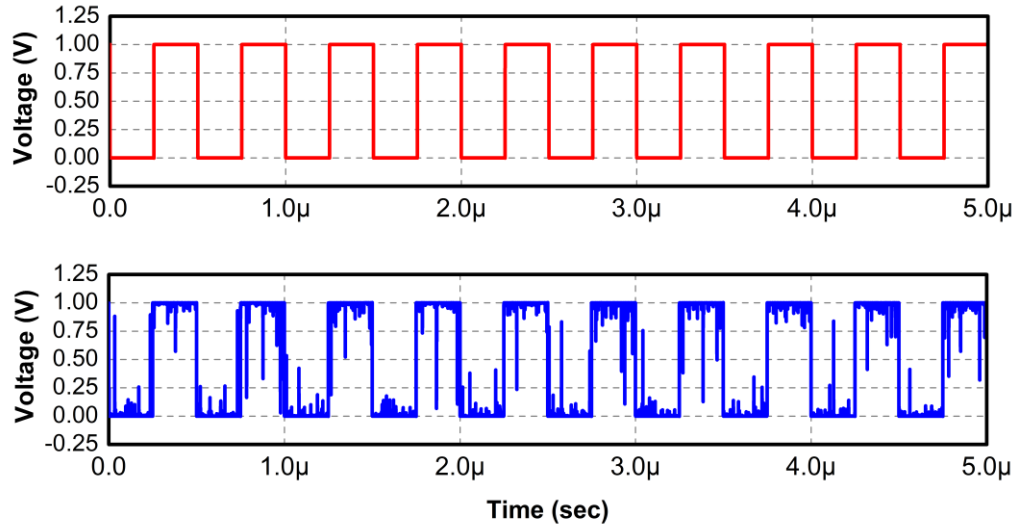


Fig 3.1 (a) The upper figure shows output signal of the CMOS inverter (b) the lower figure, as the corresponding output signal with probability of errors, p equals 0.0082.

3.2.2 Characterization of Probabilistic Behavior of CMOS Inverter

An analytical model characterizing the probability of error, p , of a non-deterministic inverter is presented for two different cases: (1) when noise is coupled to its output and (2) when noise is coupled to its input as shown in Fig. 3.2(a) and 3.2(b). The detailed behavior of an inverter with output-coupled noise is explained in Fig. 3.2(c). The digital values of '0' and '1' at the output node can be toggled incorrectly if

the noise pushes the signal above or below the switching threshold, V_m at the time of sampling, e.g. a correct output of ‘0’ is pushed over the threshold and is sampled as a ‘1’ and vice versa. The probability of this case is determined by the area under the distribution curve as represented in Fig. 3.2(c). A detailed analysis of output-coupled noise using the probabilistic approach is well explained in [53]. The probability of error is simply the sum of e_{01} and e_{10} divided by 2 since the probability of either being ‘0’ or ‘1’ is equally probable in CMOS inverter circuit. The probability of error p can then be expressed as

$$p = \frac{e_{01} + e_{10}}{2} \quad (2)$$

Expressing e_{01} and e_{10} as integrals, evaluating them and substituting the results in the above equation yields the following relationship between p , supply voltage V_{dd} , and switching threshold voltage V_m ,

$$p = \frac{1}{2} - \frac{1}{4} \operatorname{erf} \left(\frac{V_m}{\sqrt{2}\sigma} \right) - \frac{1}{4} \operatorname{erf} \left(\frac{V_{dd} - V_m}{\sqrt{2}\sigma} \right) \quad (3)$$

where erf is error function defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du$ for a real number x .

In case of input-coupled noise, analytical calculation needs to incorporate the effect of the transfer characteristics of the inverter. Since the inverter has a bias-dependent

small signal gain, peaking at the switching threshold, noise pulses with magnitude less than V_m , which would not normally result in error when coupled to the output, can be amplified and cause errors at the output. This event is explained in Fig. 3.2(d), where V_{1m} and V_{2m} are used instead of V_m due to the amplifying effect, which can be represented as Eq. (4).

$$p = \frac{1}{2} - \frac{1}{4} \operatorname{erf} \left(\frac{V_{1m}}{\sqrt{2}\sigma} \right) - \frac{1}{4} \operatorname{erf} \left(\frac{V_{dd} - V_{2m}}{\sqrt{2}\sigma} \right) \quad (4)$$

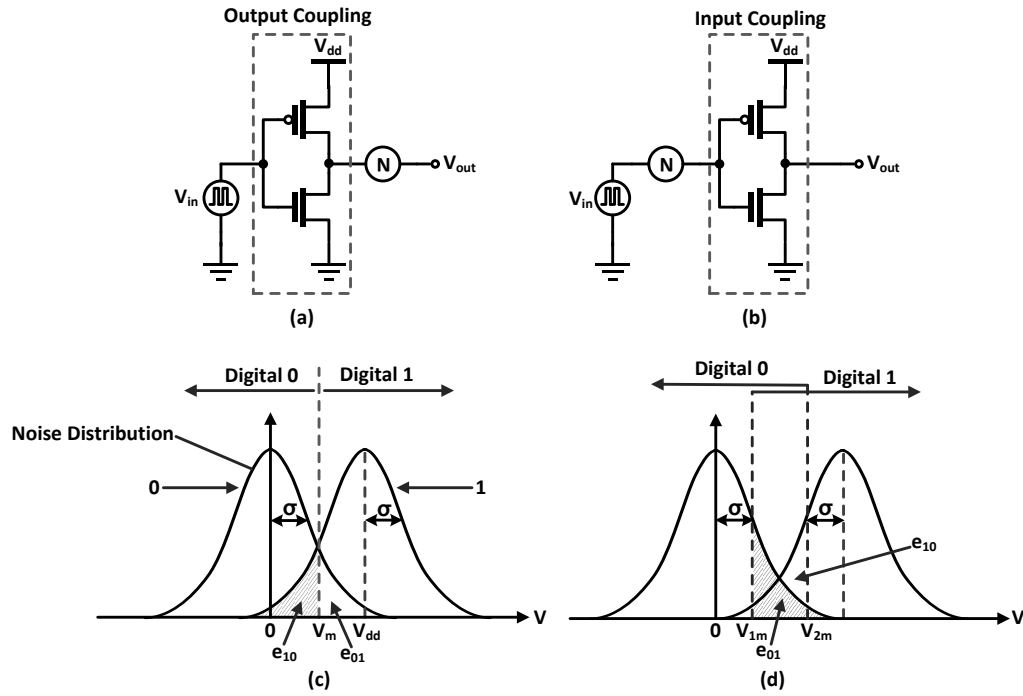


Fig 3.2. (a) Noise is coupled to the output of the inverter, (b) noise is coupled to the input of the inverter, (c) the digital value 0 (and 1) corresponding to the noisy output (input) voltage of the probabilistic inverter is represented by a Gaussian distribution with a mean value of 0 (or V_{dd}) and a standard deviation which is the rms value of the noise modeled for the input-coupled cases, (d) probability of errors from the input noise is always greater than one from the output noise because 0 to 1 transition happens at the value less than 0.5 V and 1 to 0 switching does at the value greater than 0.5 V.

In addition, even without considering the effect of amplification, the switching threshold voltage of the inverter is not always guaranteed to be half of V_{dd} due to the

pre-determined ratio of PFET and NFET gate width, process variations, temperature, etc. In this case of asymmetrical transfer characteristics, V_{1m} and V_{2m} should be used for analytical modeling rather than V_m in Eq. (3). As shown in Fig. 3.2, the error probability of input-coupled noise case would be generally greater than the case of output-coupled noise. The input noise dominance can simplify our proposed method in calculating the probability of errors and this will be verified in the later section. Note that the only noise sources considered here are random white Gaussian noise coupled to the input and the output nodes; other noise sources arising in the ground and power supplies are discussed in detail in Section 3.4.

3.3 Simulation Framework and Experimental Methodology

To validate the probabilistic approach, circuit simulations using Cadence Spectre or Synopsis HSPICE are performed utilizing process models of IBM's 45-nm Fully Depleted Silicon On-Insulator (FDSOI) device technology. Since FDSOI device is experimentally shown that process variability such as V_{th} , DIBL, and current onset voltage (COV) is well suppressed compared to bulk CMOS, this can motivate us to apply our probabilistic approach with less effect from intrinsic variability of the device itself when performing statistical analysis. Moreover, time-dependent V_{th} change due to random telegraph noise is also smaller in FDSOI MOSFETs [54], so using AWGN for the noise source can be more effective for FDSOI than bulk counterpart. Some important parameters used during the simulation are summarized in Table 1. Noise is

added to the input signal by instantiating piece-wise linear (PWL) voltage source files. The data points of the PWL source are derived from a Gaussian distribution of random numbers with zero mean and standard deviation of the noise rms value. In reality, it is not possible to fully simulate a perfect AWGN source since it would require infinite bandwidth. However, if some finite noise bandwidth is assumed and the duration of calculation time step for both circuit simulators is less than the half of maximum switching period of the PWL noise voltage sources, each data point can be regarded as a set of discrete data points sampled from a continuous AWGN function at each time step for complete recovery by Nyquist rate. This time step determines the sampling time τ [sec] of the pseudo-AWGN signal which exhibits a white noise spectral density up to a frequency of $1/\tau$ [Hz]. It is therefore important for the simulator time step to be small enough such that the noise signal has a white power spectral density up to the f_{\max} of the 45-nm SOI CMOS transistors in the circuit. 500 GHz is adequate to achieve this in the technology chosen here [55]. Our choice of $\tau = 1$ ps satisfies this requirement. The noise sources located at different nodes in the target circuits are completely uncorrelated, so each set of data points for different noise sources should be updated independently by generating new sets of random numbers. For this analysis, AWGN sources are placed at input, output, ground, and supply nodes. Input, output, and noise signals are assumed to be analog and sampled at the same frequency, so all sampled signals have identical number of data points. The error probability, p , is determined by probing the output node, and it is computed as follows.

$$p = \frac{\text{Number of samples with incorrect values}}{\text{Total number of samples}} \quad (5)$$

A switching threshold voltage of $V_{dd}/2$ will be used for p calculation. The numerator of the above expression is calculated by comparing the output samples for both the inverter with noise sources and without noise sources, as shown in Fig. 3.3. If the difference of each output voltage sampled at the same calculation time step is greater than the switching threshold, then the output at this sampling time is treated as an error. To simplify our validation approach, we only consider samples where the output levels are greater than $0.9 V_{dd}$ or less than $0.1 V_{dd}$ since any output values greater or less than those values can be thought of as being in steady-state. In other words, comparisons are made only when the output voltage from the noise-free inverter is at V_{dd} or GND. Finally, the data from these simulations are employed in MATLAB to calculate the probability of error.

Process technology	IBM 45 nm SOI	V_{dd} scaling range (V)	0.5 – 1.1
Nominal Vdd (V)	1	$\sigma_{in}, \sigma_{out}$ (V)	0.025 – 0.8
(W/L) PFET	152 nm/ 40 nm	σ_{supply} (V)	0.05 – 0.8
(W/L) NFET	209 nm / 40 nm	σ_{GND} (V)	0.05 – 0.8

Table 1. Simulation parameters for probabilistic CMOS logic circuits

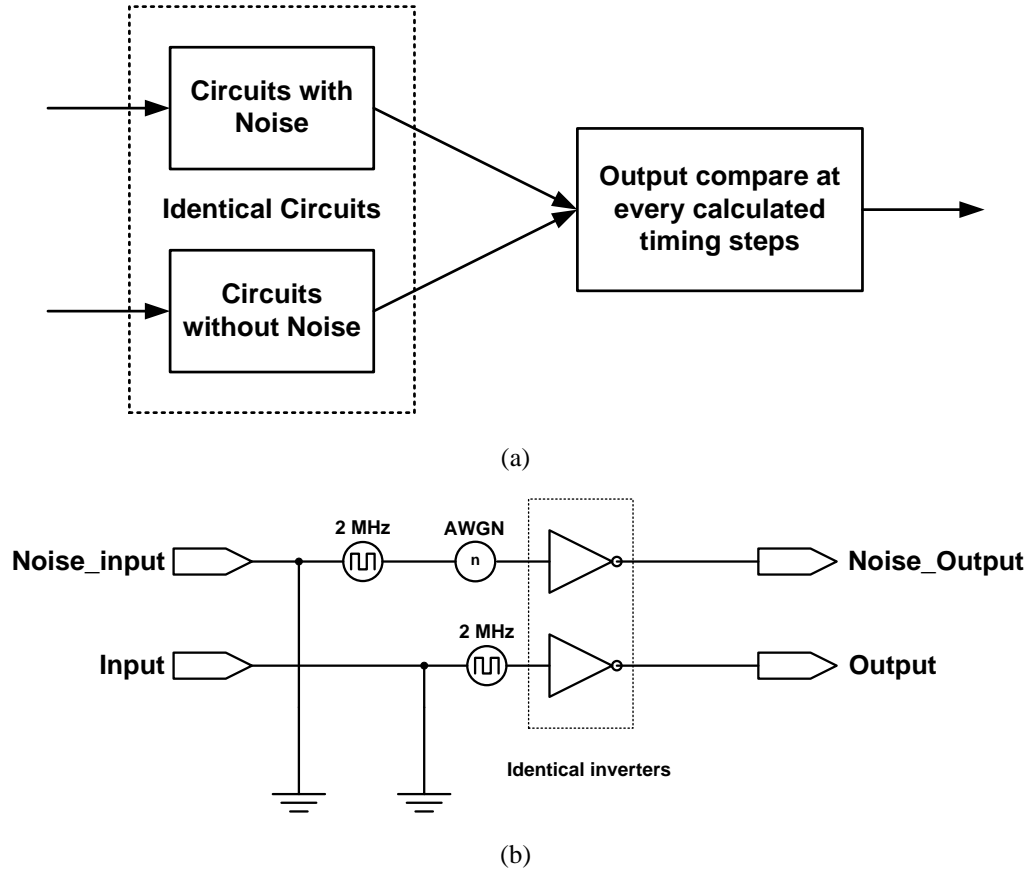


Fig 3.3 (a) Due to the difficulties imposed by the limit of randomness of AWGN and circuit simulators, a comparison method is employed for evaluating the probability of error. (b) Simulation schematic for a CMOS inverter.

3.4 Impact of Input Noise on the Probability of Error

Following our simulation as described in Fig. 3.3, various types of statistical variations, i.e., cross-talk, external noise, supply noise, or ground bounce, can be modeled using the AWGN sources which located at the different nodes in the circuit as shown in Fig. 3.4. The relationship between the error probability and the noise rms values for various types of noise sources from the simulations is shown in Fig. 3.5. Each simulation is performed to evaluate the impact of different noise sources on the probability of error at output node. All noise sources employed in this simulation are

uncorrelated and have sampling frequency of 1THz. Fig. 3.5 shows a dominant impact of input- and output-coupled noises over V_{dd} - and GND-coupled cases. To explain the differences between the input- and output-coupled noises, the probability of error for noise at two different noise sampling frequencies is plotted together with analytical approximations as shown in Fig. 3.6. In Fig. 3.6, two large discrepancies between the analytical model and simulation results are observed for the case of input-coupled noise. One is a large difference in error probability when the input-coupled noise is sampled at different frequencies, and the other is error probability due to the input-coupled noise is not negligible in the region of small rms values. To explain the large discrepancy between the analytical model of the input-coupled noise and the simulated results, we consider the AC transfer characteristics of the inverter and high frequency properties of the device used for this simulation. Only two different values of DC biases are considered because we excluded the output values in the transition region for this analysis. Both the NFET and PFET show an increasing cut-off frequency with increasing DC bias, as expected. A high-frequency simulation shows that the NFET has cut-off frequencies of 314.5GHz and 3.953GHz when biased at 1V and 0V, respectively, while for the PFET these values are 247.5GHz and 0.85GHz. In the simple analytical model of the input-coupled noise, this frequency-dependent transfer characteristic is not considered at all. This omission is the main cause for the discrepancy between the analytical model and the simulated data as the transfer characteristic causes an effective filtering of input noise. This explanation becomes clearer when we compare the two plots of error probability sampled at 1THz and 1GHz. With 1GHz sampling, a very small part of input noise signal is cut-off.

However, in case of 1THz sampling, most of the noise is filtered out. In addition to high-frequency properties of inverter, AC transfer characteristics of the inverter circuit are introduced to explain why input-coupled noise is dominant especially in the region of smaller noise rms values. An AC signal gain of a simulated inverter circuit is plotted as a function of DC bias values in Fig. 3.7. According to this plot, when the inverter circuit is biased at both V_{dd} and GND, signal gain is so small that amplified input-coupled noise cannot alter its output level adversely. However, during the signal transitioning from high to low and low to high, bias voltages are changed, so AC signal gain is increased enough to translate an input noise to wrong output. This amplification of noise at different biases is displayed in Fig. 3.8. Amplified signals during the transition are detected as an error even in case of relatively small noise rms values. Fig. 3.9 shows how the amplified noise can change the timing accuracy.

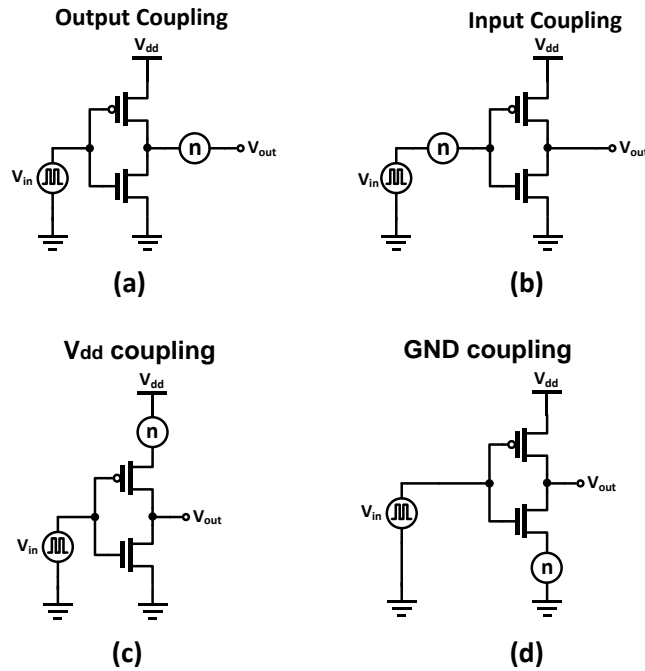


Fig 3.4. Models of various types of statistical variations in CMOS inverter (a) noise from the following stages (b) input noise from external sources and cross-talk (c) supply noise and V_{dd} droop (d) ground and substrate noise

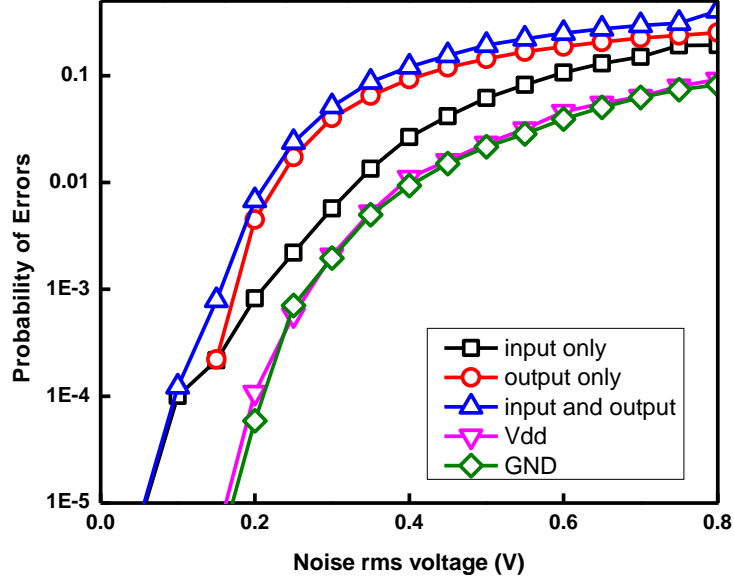


Fig. 3.5. Probability of error vs. noise rms values with various types of noise sources, input-coupled only, output-coupled only, input/output-coupled, Vdd-coupled, and GND-coupled. All noise sources are uncorrelated and sampled at 1THz. See Figure 6 for comparison with 1GHz sampling.

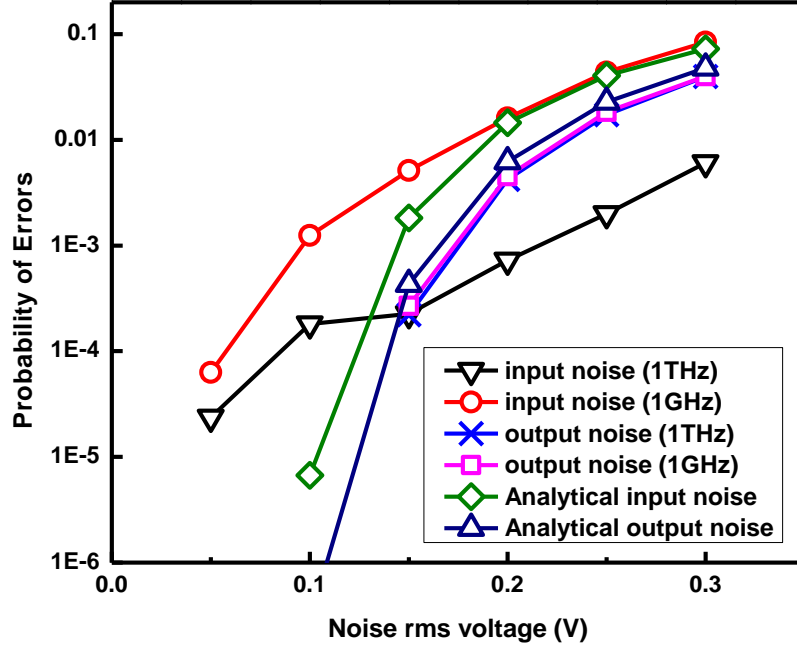


Fig. 3.6. Probability of error vs. noise rms values with two different sampling frequencies and the analytical modeling introduced in Section 2. Sampling frequencies are 1THz and 1GHz, respectively and probability of error from the analytical modeling is evaluated by integrating the corresponding regions of the graphs shown in Fig. 2. Due to the frequency-dependent transfer characteristic of the inverter, output-coupled noises with different sampling frequencies show the same probability of error. Furthermore, the previous analytical model of output-coupled case estimates the result of simulation very closely

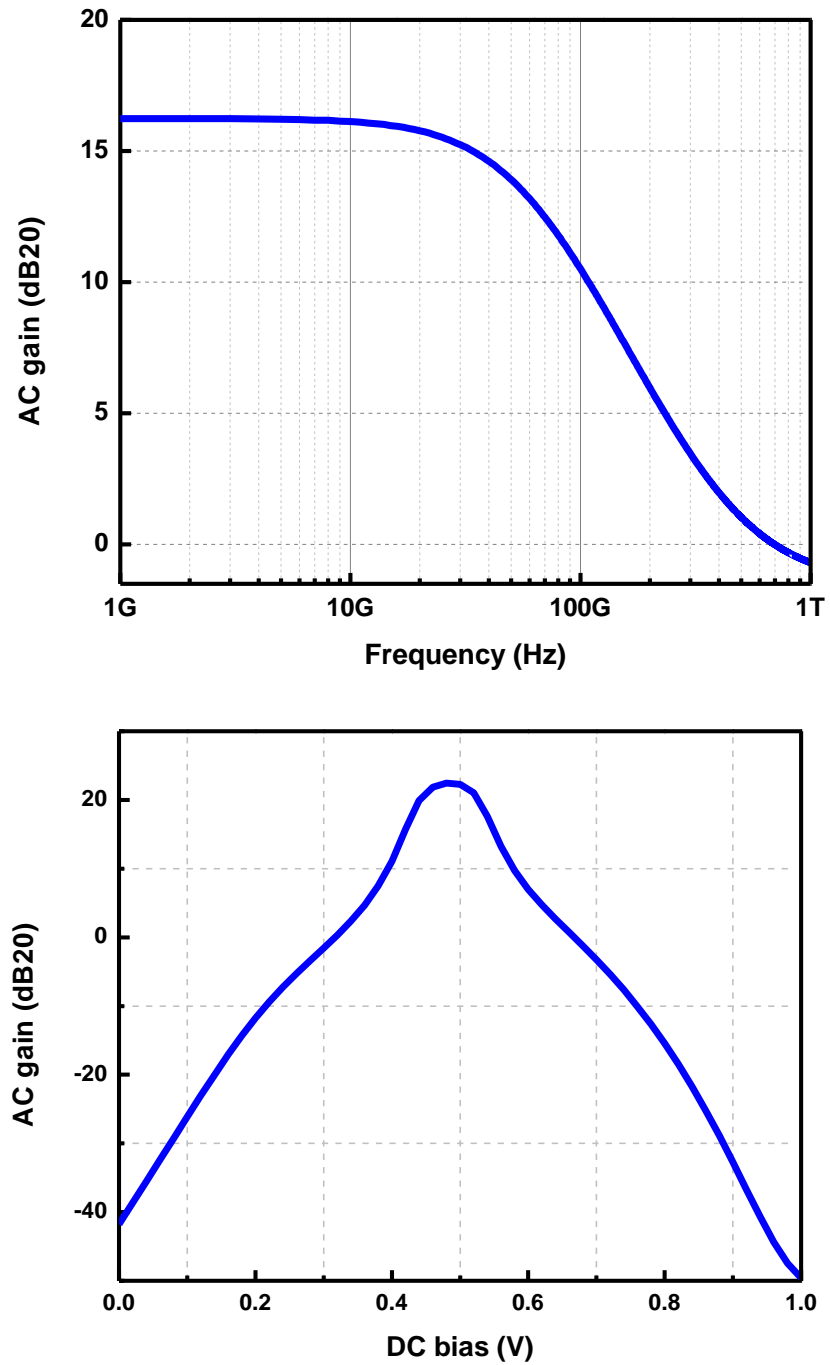


Fig 3.7 AC gain and frequency response plot of unit-sized inverter using IBM's 45 nm SOI CMOS process. AC gain is plotted as a function of DC bias values at 1GHz of input sinusoid and frequency response of inverter is swept from 1GHz to 1THz at the DC bias of 500 mV. Note that over 500GHz, amplification effect of the inverter is diminished, this effect also contributes the less probability of errors in the region of low rms values.

In synchronous digital systems, there are two types of errors that affect the performance of a target system. One is amplitude noise (or glitch) and the other is timing error, also known as jitter. Additional circuits, such as self-timed gating approach, eliminate the glitch propagation [56], however, critical path delay should be increased by two times of maximum jitter to allow worst case timing variations. Since jitter directly affects the performance of a sequential digital system, jitter is one of the most dominant effects worsening the performance of a system. As a consequence, this input-coupled noise effect dominates over output-coupled, supply voltage-coupled, and ground noise as shown in Fig. 3.5. Due to this impact of an input noise and frequency response, all simulations in the rest of the manuscript are performed with only input-coupled noise with 1GHz sampling and this dominance makes the simulation simpler and faster.

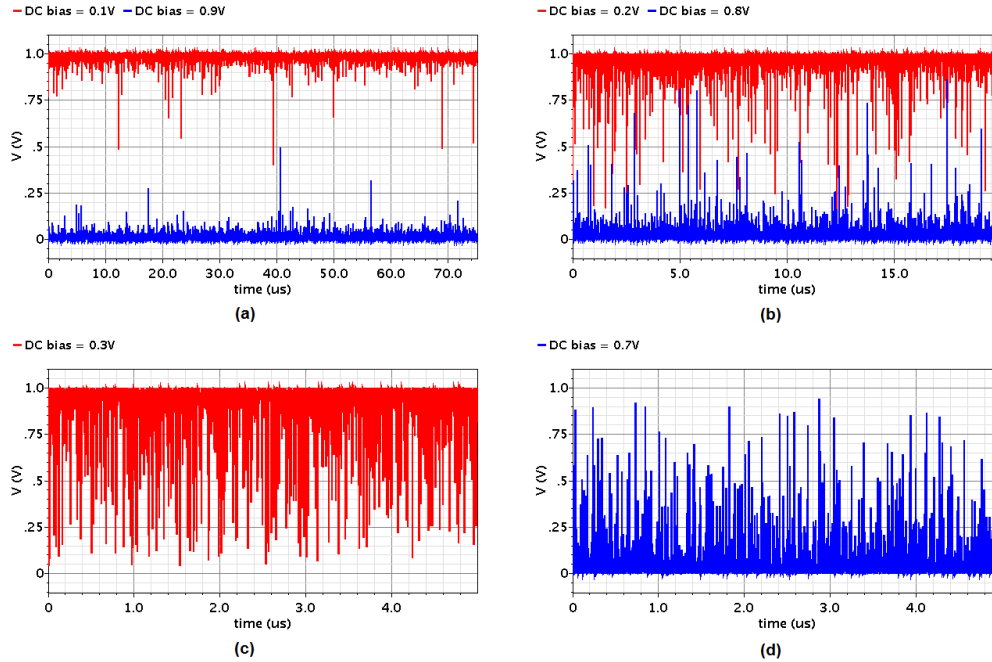


Fig 3.8. Noise amplification effect at different biases, $0.1 V_{rms}$ noise is amplified due to transfer characteristic of the inverter. (a) shows a small number of erroneous output observed even at DC biases of 0.1 V and 0.9 V (b) shows at DC biases of 0.2 V and 0.8 V (c) at 0.3 V DC bias, and (d) at 0.7 V DC bias

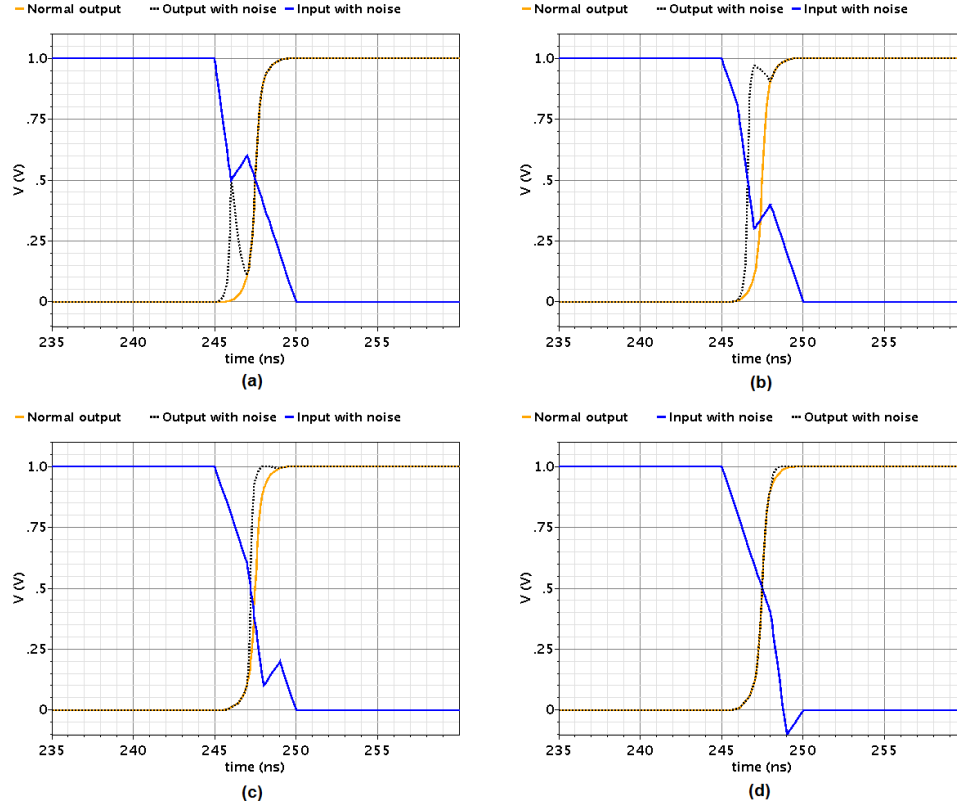


Fig 3.9. Input noise translates into jitter during input transition. Note that the noise near the switching threshold ($V_{dd}/2$) generates glitch as well as timing error as shown in (a) and (b). On the other hand, noises near V_{dd} or GND are not translated into the jitter as in the case of (d).

To demonstrate how this dominance of input-coupled noise extends to larger systems, we implement 1-bit full adder circuit composed of 2 XOR gates and 3 NAND gates. Two simulations for evaluating the error probability are performed. The first places three uncorrelated additive white Gaussian noise sources on inputs, A, B, and C_{in} , while the second places noise sources at the gates of the three internal nodes and two output nodes as well as the input noise sources. This scheme is shown in Fig. 3.10. Simulation results show that the error probability of the output C_{out} is almost the same for both cases, for example, with 0.1 V_{rms} value of additive white Gaussian noise sources, a probability of error for the case of only input-coupled noise is 0.001329 and

0.001274 for the other case. This supports the dominance of input-coupled noise extending to the case of a larger system. In addition, with the input-noise dominance approach, computing resources and elapsed time for simulation decrease by 8% and 12%, respectively. If this simplifying method is applied to even larger system such as 4-bit CCS-CSS adder block introduced in later section, the reduction in simulation cost will increase further.

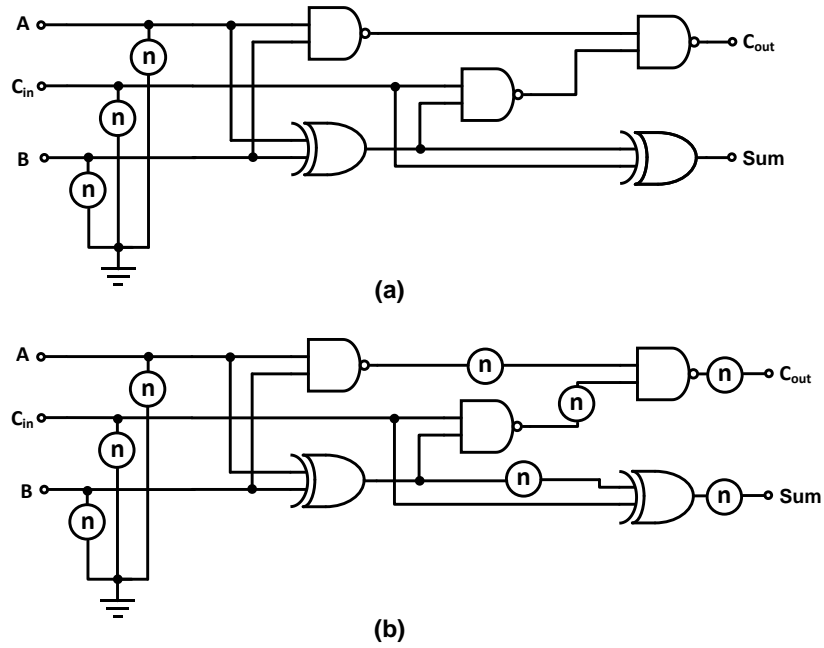


Fig 3.10. Simulation scheme for 1-bit full-adder circuit. In (a) 3 independent noise sources are located only at input nodes. (b) incorporates 8 independent noise sources at every node.

3.5 Error-Energy Relationship for Gate-Level Logic Implementation

Starting from an inverter as a foundational element of non-deterministic CMOS logic, our simulation methodology is extended to simple gate-level implementation of

this probabilistic circuit. Simulation is performed to reveal a relation between energy per logic operation and probability of error. Employing inverter, 2-input NAND, and 2-input XOR with input-coupled noise at 1GHz sampling of 30 mV and 100 mV rms values, a framework for inexact calculation scheme is established and verified. In Fig. 3.11, energy vs. error relations for inverter, NAND, and XOR are plotted for noise rms values of 30 mV and 100 mV, respectively. The error relationship to energy is exponentially related through Boltzmann relationship as a simple example of statistical equivalence between thermodynamics and information. This probability of correct information processing can be related to the energy of the corresponding logic operation as

$$p_{calc} \propto \exp\left(-\frac{\Delta E_{calc}}{kT}\right) \quad (6)$$

where k is Boltzmann's constant and T is absolute temperature. For an inverter, e.g., a similar relationship is articulated in Mead and Conway [57]. If we substitute $1-p$ into p_{calc} , then

$$1 - p \propto \exp\left(-\frac{\Delta E_{calc}}{kT}\right) \quad (7)$$

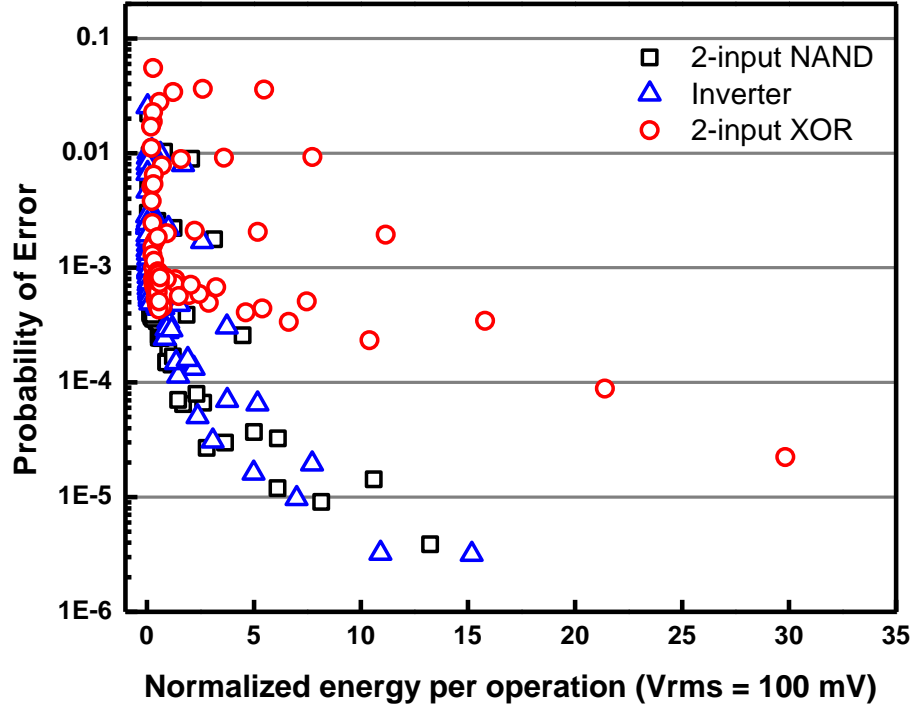
leading to

$$\log_e(p) \propto -\frac{E_{logic}}{kT} \quad (8)$$

or

$$\log_{10}(p) \propto -\frac{0.434}{kT} \cdot E_{logic} \quad (9)$$

As stated in Eq. 9, our simulation reveals an interesting relationship between errors and energy. The ordinate in Fig. 3.11, which is logarithm with base 10 of errors, has a linear dependency on the energy of logical operation. This observation simply follows the statistical relationship from thermodynamics.



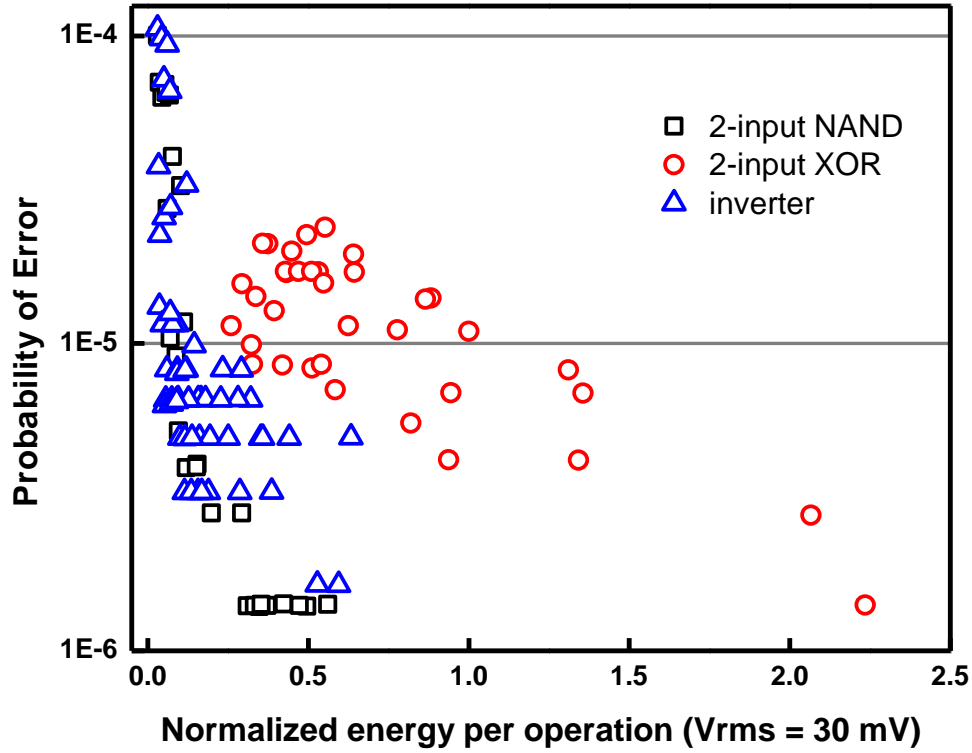


Fig 3.11. Relationship of energy per bit operation vs. probability of error for inverter, NAND, and XOR. Simulation is performed with noise of V_{rms} value of 30 mV and 100 mV, respectively. Supply voltages and threshold voltages of the gate-level circuits are varied for obtaining different values of power consumption. Energy per bit operation is calculated by multiplying each power consumption values by minimum delay propagation time.

In reality, noise rms value of kT/q ($\sim 25mV$) looks reasonable for this type of error calculation. However, employing this value, the probability of error would be too small to be detected by the simulation tools such as SPECTRE and HSPICE. For example, from the Boltzman distribution and degree of freedom principle, error rate of the single circuit element can be estimated to 10^{-23} [58]. To detect this error, even with one instance of error, the required simulation sample should be greater than 10^{23} . This is practically impossible within the current computing environment. This is why the noise rms value is exaggerated to be 4 times greater than the realistic value. As seen in

Fig. 3.11, most of the data point is saturated by this limitation and still stays at the vicinity of zero power. Despite of this discrepancy, some data points in Fig. 3.11 clearly shows the linear dependency of the logarithmic error probability on the energy.

3.6 Power Savings via Inexact Computing

3.6.1 MSB-LSB Weighted Scaling of Supply voltages

When performing computation, errors in the most significant bit (MSB) position will produce larger calculation errors compared to errors in the least significant bit (LSB) position as shown in Fig. 3.12. Based on this observation, a specialized adder architecture is proposed in this section to exploit the benefits of inexact computing.

$$\begin{array}{ccc}
 \begin{array}{r}
 \boxed{0}1100111 \\
 \times \boxed{00}010011 \\
 \hline
 1,957
 \end{array}
 & \xleftarrow{\text{Errors in MSB}}
 \begin{array}{r}
 11100111 \\
 \times 01010011 \\
 \hline
 19,671
 \end{array}
 & \xrightarrow{\text{Errors in LSB}}
 \begin{array}{r}
 1110\boxed{1}111 \\
 \times 0101001\boxed{0} \\
 \hline
 19,598
 \end{array} \\
 0.901 & \text{Calculation Error} & 3.725\text{E-}03
 \end{array}$$

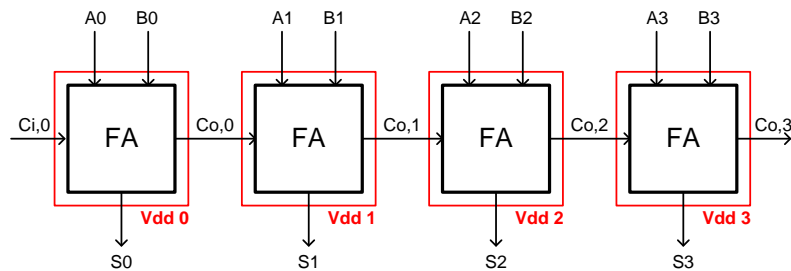


Fig 3.12. Errors in MSB position produce larger calculation errors than errors in LSB. This observation leads to the so called *MSB-LSB weighted supply scaling* scheme as an example of ultra low-power computing system

3.6.2 Architecture of Adder

While lowering the supply voltage reduces the dynamic power consumption quadratically, it also results in significant performance degradation. Therefore, if a better energy-efficiency is required within the constraints of bounded error rate, based on the relationship between individual bit error and total computation error, it would be wise to reduce the supply voltage of the LSB blocks. A high supply voltage is still recommended for the MSB blocks for more accurate calculation. With this thought, a 32-bit CCS-CSS (Conditional Carry Select-Conditional Sum Select) adder composed of 8 identical sub-blocks of 4-bit CCS adders as described in Fig. 3.13 was designed. The CCS adder pre-generates sum and carry-out without a carry-in value propagated from the previous block, and these pre-calculated values are multiplexed by the carry-in value when it becomes available. Compared to other adder architectures, the CCS adder determines the outputs dependent solely on the input vectors of the current block. Even though the sum and carry-out of the current block are eventually determined by the propagated signal generated from the previous block, the pre-generated values without carry-in are evaluated with higher supply voltages. Therefore, the calculation accuracy of the current block is less susceptible to calculation errors from the previous block compared to other adder architectures. This is the reason why CCS architecture is chosen for a demonstration of this MSB-LSB weighted supply voltage scheme in this section. Moreover, critical path delays in this adder architecture are the paths for calculating the MSB blocks. This means that the LSB blocks are mostly non-critical paths and supply voltages for LSB blocks can be lowered than the supply voltages of

MSB blocks, in general.

A 4-bit CCS adder is implemented as shown in Fig. 3.14(a). A 2-to-1 MUX is designed using transmission-gate logic and other gate primitives are implemented with static circuits to prevent a series cascade of transmission-gates which would increase delay quadratically to the number of gates. Implementation using transmission-gate MUX is intended for layout area saving and faster operating speed. Since errors in CMOS circuits are mainly resulted from a transfer characteristic of the circuits as described in section 3.4, the input noise transferred by transmission gates can be negligible compared to other static-style CMOS gates. Simulation shows that the errors in static CMOS-style MUX is approximately 100 times larger than the transmission-gate type MUX. In our adder, this MUX is assumed to be transferring the noise to the next stage without affecting significantly. If static CMOS-style MUX was hired for this architecture, an overall error would have been increased. The overall error would be different depending on the number of stages, circuit types, and signal paths. Simulation in the later section will clarify this in detail.

Carry look-ahead (CLA) as well as block CLA are also implemented using transmission-gate MUX. Additional static buffer circuits are inserted between them to again prevent back-to-back transmission-gates. Fig. 3.14(b) shows an implementation of the CSS adder. This adder sums an incoming 1-bit carry-in to the sum value generated from the 4-bit CCS adder block. Depending on the value of the incoming carry-in signal, one of the two outputs, with and without carry-in bit, is propagated to the next block. The adder is implemented using IBM's 45-nm CMOS FDSOI process

technology. Simulated critical path delay without parasitic extraction is 90 ps and circuit is designed to operate up to 4 GHz.

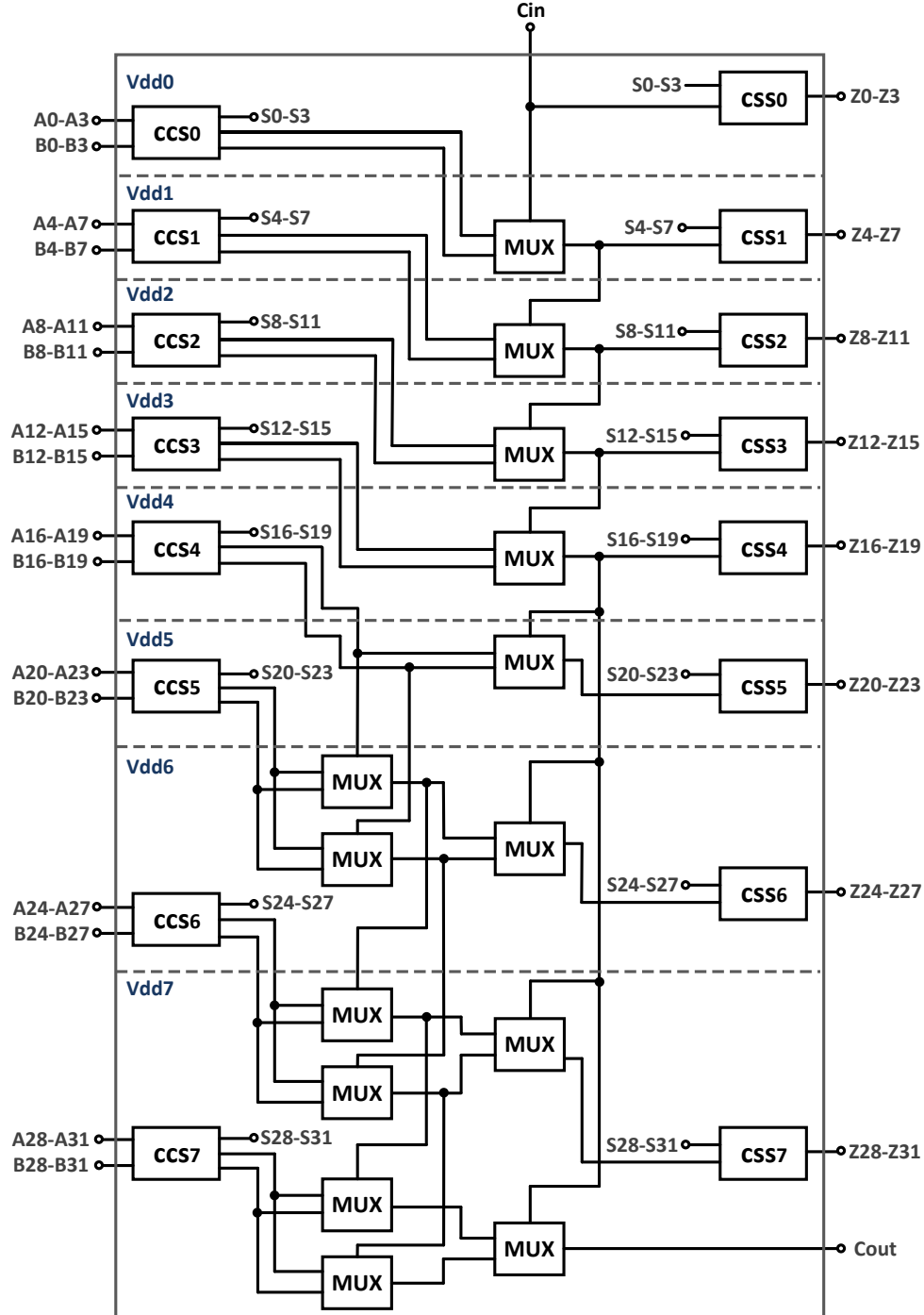
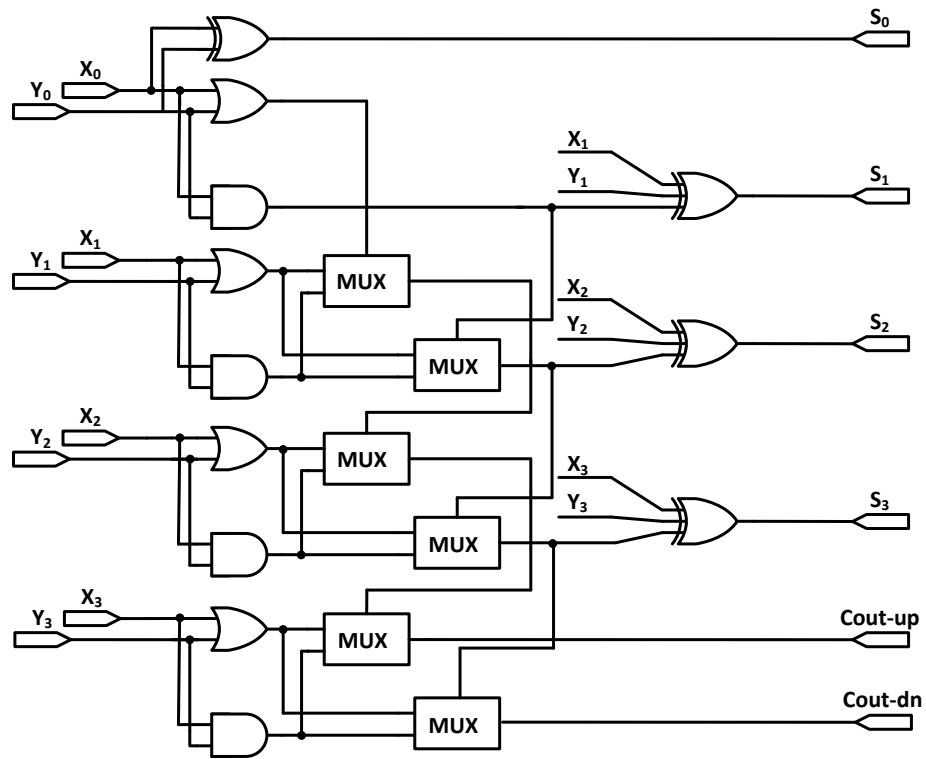
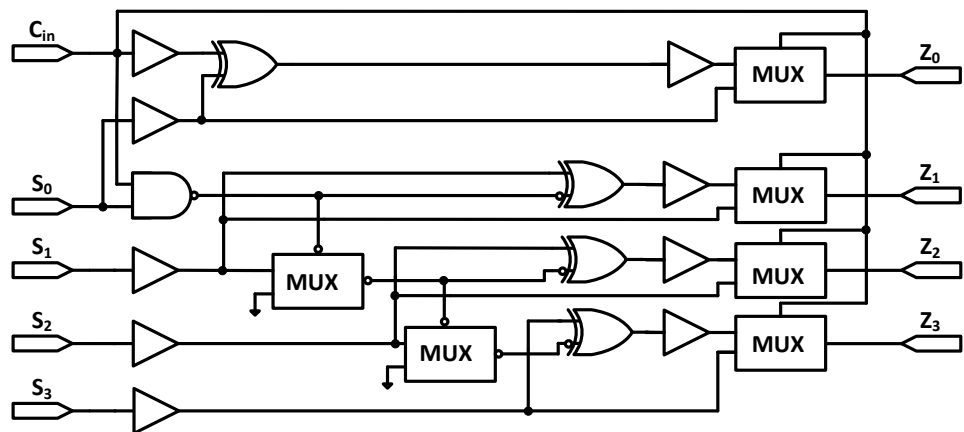


Fig 3.13. Implementation of MSB-LSB weighted scaling of supply voltages for 32-bit CCS-CSS adder.



(a)



(b)

Fig 3.14. (a) Logic implementation of 4-bit Conditional Carry Select adder block (b) 4-bit Conditional Sum Select adder block

3.6.3 Simulation Results

To validate the circuit functionality and robustness, randomly chosen 32-bit input vectors are generated and employed. The adder circuit is simulated using the Cadence Virtuoso Accelerated Parallel Simulator to expedite simulation while maintaining accuracy. The circuit is inserted between a pipelined stage clocked at 3 GHz, which allows a 30% operating safety margin for the worst case combinations of extreme variations and noise. For purposes of comparison, we select the case of a single nominal supply voltage (1 V) to be the baseline, i.e., free of computation errors. The overall power (energy) consumption for this baseline condition is defined as 100% and results for other cases are normalized to this value. Various combinations of MSB-LSB supply voltage values are selected as shown in Fig. 3.15 to find a relationship between energy vs. calculation error. For this error calculation, the adder operates at fixed clock frequency of 3GHz even in case of lowered supply voltages for the corresponding LSB blocks. Our concern is a relationship between the calculation error vs. energy consumption, at the same circuit performance. Therefore, we don't care any performance degradation from the lowered V_{dd} . This performance degradation is already taken into account as increased calculation error. Only thing we consider is how much power can be saved within some error tolerance. The results plotted in Fig. 3.16 demonstrate the effectiveness of this MSB-LSB weighted scheme. For example, if the calculation error is tolerable up to 1×10^{-6} , which is one pixel error per frame of High Definition (1280 x 720) video streaming for handheld media player, a reduction in power consumption of more than 40% is possible with this scheme.

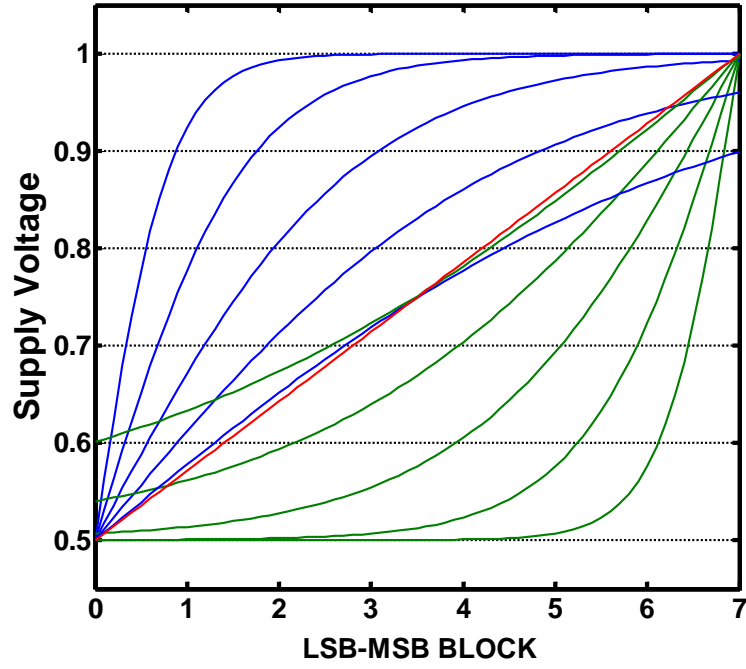


Fig 3.15. MSB-LSB bit selection map for voltage scaling. Supply voltage (y-axis) has a voltage range from 0.5 to 1 V. To have the independent supply voltages, a 32-bit CCS-CSS adder is divided into 8 identical blocks which labeled from ‘0’ for LSB (0 – 3th bit) to ‘7’ for MSB (28th – 31th bit), x-axis. For example, using the top blue line for V_{dd} selection, 0.5V is assigned to block 0 (LSB), 0.924V for block1, 0.993V for block 2, and 1V for remaining blocks (MSB).

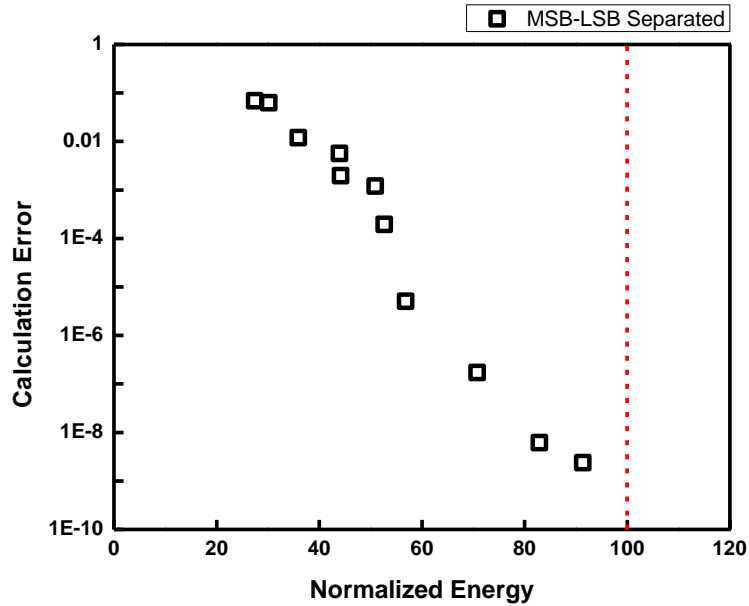


Fig 3.16. Energy vs. calculation error for 32-bit adder by applying MSB-LSB weighted scheme. Even with lowered values of supply voltages, output is sampled at 3 GHz. The difference of sampled output value and the correct value is divided by the correct value, and this ratio is referred as calculation error in this section.

$$\text{Calculation error} = \text{ABS}(\text{SUM}_{\text{error}} - \text{SUM}_{\text{correct}}) / \text{SUM}_{\text{correct}}$$

3.6.4 Ultra Low-power Data-path circuit Design Methodology

using Probabilistic Circuit

In our statistical performance metric, evaluating the relationship of error and energy for this specialized adder architecture requires 65 uncorrelated independent input noise sources and significant amount of transient-response simulation time to collect enough number of data samples for calculating the error probability. This is very time consuming and perhaps practically impossible. To address the difficulty of complex simulations, a simplified approach is employed. As shown in Fig. 3.13, each 4-bit adder in the divided blocks has almost same architecture; therefore, if we characterize the statistical performance of each block, a simplified approach used for the basic building blocks can be extended to entire architecture without loss of consistency. Noting that outputs from each block contribute to the final calculation output value, the weighted sum of probability of error in each bit is approximately the total calculation error. Assuming $p_{ezi}(V_j)$ is the probability of error in i^{th} bit of j^{th} 4-bit adder block, the total calculation error and power consumption are

$$\text{Error} = \left[\sum_{i=0}^3 2^i p_{ezi}(V_1) + \sum_{i=4}^7 2^i p_{ezi}(V_2) + \cdots + \sum_{i=28}^{31} 2^i p_{ezi}(V_8) + 2^{32} p_{ecout}(V_8) \right] \\ \div 2^{32} \quad (10)$$

with

$$\text{Total Power consumption} = \sum_{j=1}^8 P(V_j) \quad (11)$$

For simplicity, carry-in bit is being treated as another independent input-coupled noise source rather than evaluating the probability of error in each block based on the actual carry-in bit from the previous block. The result of full-error simulation of 4-bit adder block is displayed in Fig. 3.17. The probability of error in each output bit is exponentially decreasing as the supply voltage of each block increases. The probability of errors in each bit is different because each output bit has different number of propagating stages and signal paths. Using the probability of error from the basic building block, we can find the optimized set of supply voltages values given to each block while minimizing calculation error with the constraint of power budget. Inversely, with the bounded limit of calculation error, we can find the optimized set of the supply voltages to have the minimized power consumption. Fig. 3.18 shows a result from this optimization methodology based on Eq. 10 and 11. In this calculation, supply voltages are varied from 0.5 V to 1.1 V with an increment of 100 mV. Compared to conventional DVS (Dynamic Voltage Scaling), simulation result using a commercial 45-nm CMOS technology indicates that this MSB-LSB weighted scheme can provide 250% higher energy-efficiency for the same bounded error rate, or improve 10^4 times error robustness for the same power consumption.

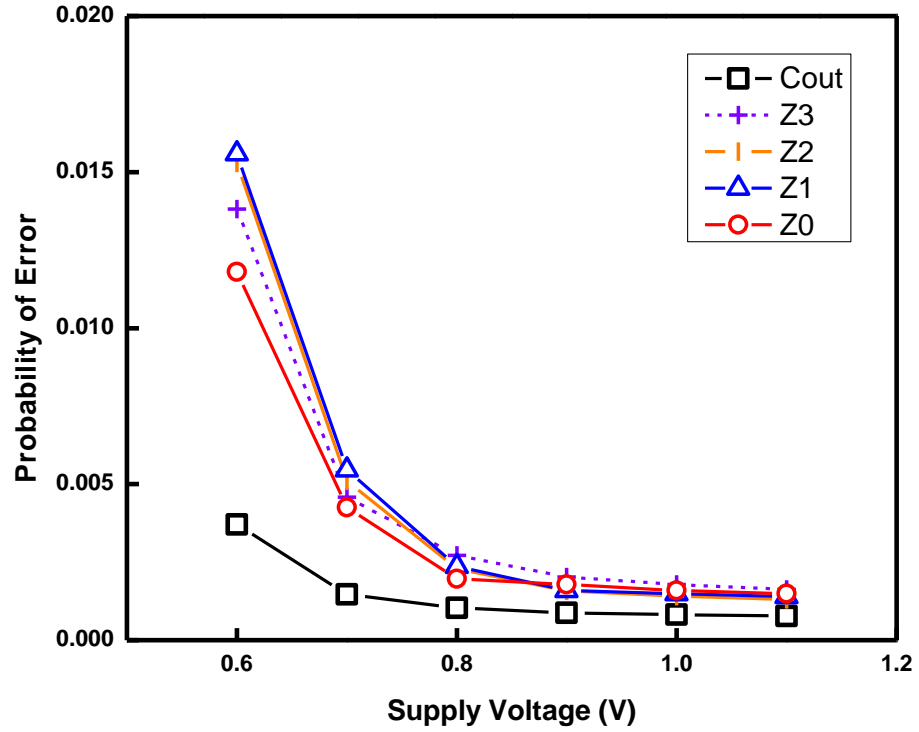


Fig 3.17. Probability of error for a 4-bit CCS-CSS adder block.

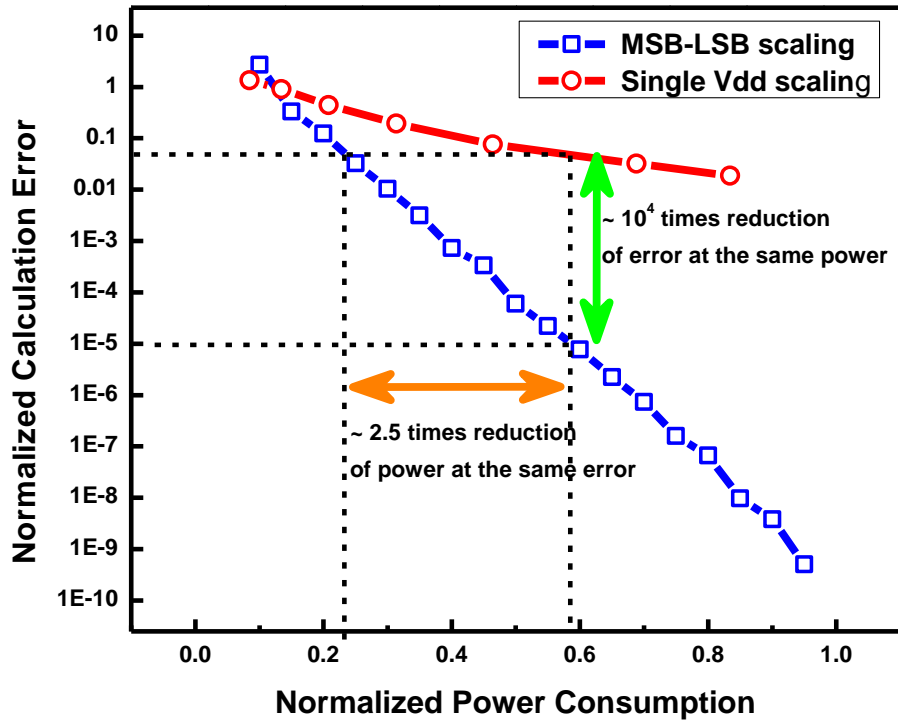


Fig 3.18. Power vs. Calculation error based on the optimization methodology suggested by Eq. 10 and 11. Note that this plot is exactly same as the plot in Fig. 16 and verifies our optimizing methodology.

In previous section, our probabilistic approach using gate-level implementation of CMOS circuits shows a clear governance of 2nd law of thermodynamics in terms of error vs. energy. Similar governance is shown in larger circuit-level. Comparison of Fig. 3.16 and 3.18 reveals an exact agreement between the calculation error of adder simulation result in section 3.6.2 and the error probability calculation in section 3.6.3. The first simulation is performed to evaluate the actual calculation error. Randomly generated digital input vectors are applied to the adder and each V_{dd} value for 4-bit adder block is selected from the graphs in Fig. 3.15. All outputs are sampled at fixed frequency of 3GHz and the calculation errors are evaluated by comparing with the correct calculation value. Some outputs will have calculation errors since its LSB block is biased at lower V_{dd} and can't calculate correctly at 3GHz. Then pairs of calculation error and energy values are scattered in Fig. 3.16. The later one is performed based on the optimization of Eq. 11 and 12, and combined with the probability of error from the suggested simulation framework. At the energy constraint of 10% to 100%, optimized V_{dd} sets are found to have the minimum error probability. This agreement confirms our probabilistic approach is valid and still following the governance of 2nd law of thermodynamics even with some assumptions and simplifications.

Other approaches such as CVS (Clustered Voltage Scaling) [59] and using multiple supply voltages [60] could be also employed in the implementation of this MSB-LSB weighted ALU. However, our proposed scheme does not require that the critical-path blocks be separated from the non-critical-path blocks, thus adding minimal extra routing for each supply voltage. Since a typical chip floor plan is inherently

parallelized by the blocks which are already divided by output bit, there is no need to re-design the entire ALU block for this scheme. In addition, the proposed scheme allows easy configurability to meet a wide range of specifications – from high computational accuracy to ultra low power computation – simply by adjusting the supply voltage value. However, employing a number of independent voltage sources increases the area overhead of input pads for off-chip sources or the design overhead for on-chip voltage controller. Intuitively, more voltage sources give improved energy-efficiency, however, also increase design overhead and implementation cost. Fig. 3.19 indicates a trade-off between the numbers of independent voltages sources vs. energy-efficiency of our MSB-LSB weighted scaling scheme. As a final comment, full-custom ASIC chip implementation would be the best testing and validation approach for this ALU scheme, however, a quick prototype using eight identical FPGA chips each with a separate supply voltage would also be adequate in terms of demonstrating the merits of the proposed computational scheme.

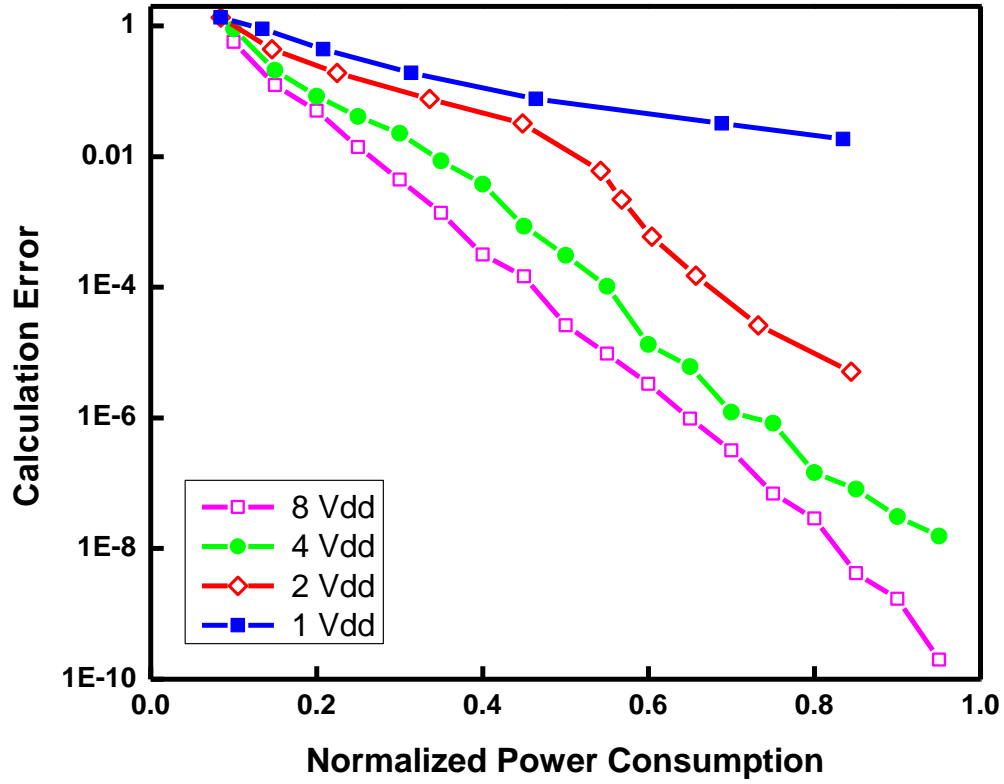


Fig 3.19. Power vs. Calculation error for the different numbers of independent voltage sources. Note that even with double or quad number of voltage sources, MSB-LSB weighted supply scaling method is still very powerful in terms of energy-efficiency.

We elaborated the *Inexact Computing* approach and its simulation framework, and extended this methodology to design of ALU for the first time. Due to its uniqueness, it may look harder to compare directly with conventional methodologies. However, novelty of this work still can be viewed using two approaches separately, multiple supply design and error tolerable design, respectively. Multiple V_{dd} design has been researched as an effective way to save the power without sacrificing the performance. For example, a high-performance 64-bit ALU was designed using dual supply and reported up to 22% of power saving [61]. The advantage of MSB-LSB scaling over the conventional multiple supply design is already discussed, so the design aspect in

terms of error tolerance will be highlighted here. Recently, many researchers have investigated a new approach – Aggressive Deployment [62] – as a method to completely eliminate a safety margin in adaptive circuit design. This idea is mainly based on the following observation, “Current designs target worst-case conditions, which are rarely encountered in actual system, so operate circuits at lower voltage levels than allowed by worst case, and deal with the occasional errors in other ways.” As a first example using single-rail DVS, 18×18 multiplier implemented on an FPGA showed a 1.3% error rate for power reduction of 35% without any correction algorithm [63]. The recent RAZOR II architecture [64] showed 30% power saving with 7% of area overhead for error detection and correction. However, their implementation requires a significant amount of additional clock energy and the error correction circuits are susceptible to metastability and have a substantial design overhead. Employing our *Inexact Computing* approach, power savings can be achieved and the extra cost to detect and correct errors or inserting safety margins to guarantee correct functionality can be avoided for applications where a low level of error can be tolerated.

3.7 Chapter Summary

In this work, an approach to low-energy computation using statistical performance metrics that incorporate error margins as a constraining requirement is elaborated. A probabilistic approach allows one to explore the implications. Using sub 50-nm device technologies where variations start increasing, the approach is employed and

demonstrated in computation. A detailed analysis of transfer characteristics of the inverter is employed to explain the effect of input-coupled noise and the dependence on sampling frequencies. Simulation results show that the input-coupled noise dominates the total noise at the output and is significant in determining the probability of error; this dominance is demonstrated in more complex circuits as well. Utilizing the probability of error as an allowable design tolerance, we show that a simultaneous optimization of both energy-efficiency and computing error is possible. This provides the circuit designer with greatly increased flexibility to trade off between energy and calculation accuracy. As an application of this concept, a 32-bit MSB-LSB weighted supply voltage scaled adder with carry look-ahead capability is implemented to show the potential benefits of inexact computing. Our simulation results using a 45 nm CMOS SOI technology indicate that this new adder architecture can reduce the total power consumption by more than 40 % while resulting in a calculation error of only 10^{-6} .

CHAPTER 4

ULTRA-LOW POWER ALU AND DSP CORE FOR INEXACT COMPUTING

4.1 Motivation and Background

In the previous chapter, we established a novel simulation framework using the proposed probabilistic representation of digital CMOS circuits. In addition, a 32-bit CCS-CSS adder architecture was demonstrated to show the improved energy-efficiency at the cost of calculation accuracy, or vice versa. However, in numerous applications where the inexact computing methodology can be applied due to the bounded error tolerance, the actual computing workhorse is generally not a simple adder circuit. Since our previous verification methodologies are still relatively simple from a hardware standpoint, more generalized verification is required to extend the concept of inexact computing to real world applications.

A general digital processor consists of data-path, memory, control, and input/output blocks. The data-path is the core of the processor – this is where all computations are executed. The other blocks in the processor are support units that either store the results produced by the data-path or help to prioritize tasks in the next clock cycle. A typical data-path is composed of a large number of basic combinational functional blocks, such as arithmetic operators (addition, multiplier, comparison, and shift) or logic (AND, OR, and XOR) operators. Of these combinational logic blocks, adders and multipliers are the most complex and have the greatest overall effect on the power consumption and performance of the entire digital processors.

There is another specific type of digital processor known as a Digital Signal Processor (DSP), which has an architecture optimized for the operational needs of digital signal processing. Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repeatedly on a series of data samples. Signals (perhaps from audio or video sources) are constantly converted from analog to digital, manipulated digitally, and then converted back to analog form. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within some fixed time, and deferred (or batch) processing is not viable. Most general-purpose microprocessors can execute DSP algorithms successfully, but are not suitable for use in portable devices such as cell phones and smart phones because of power supply and space constraints. A specialized digital signal processor, however, tends to provide a lower-cost solution with better performance, lower latency, and no requirements for specialized cooling or large batteries.

In digital signal processing, the multiply–accumulate operation is the most common step that computes the product of two numbers and adds that product to an accumulator. The hardware unit that performs the operation is known as a multiplier–accumulator (MAC, or MAC unit); the operation itself is also often called a MAC or a MAC operation. Digital Signal Processors contain a dedicated MAC, consisting of a multiplier implemented in combinational logic followed by an adder and an accumulator register that stores the result. The output of the register is fed back to one input of the adder, so that with each clock cycle, the output of the multiplier is added to the register. Combinational multipliers require a large amount of logic, but can

compute a product much more quickly than the method of shifting and adding typical of earlier computers.

Due to the importance of this operation, the design of energy-efficient and high-performance adders and multipliers is crucial for modern general-purpose processors as well as digital signal processors. This is why the remaining content of this chapter is focused on the design methodology of energy-efficient ALU building blocks optimized using the method of MSB-LSB weighted supply scaling. As verified in the previous chapter, this approach results in better calculation accuracy for the same energy, or vice versa.

Computing applications suitable for inexact computing mainly rely on digital signal processing. Some examples include the human vision-audio system, wireless communication via noisy channels, encoding and decoding of bit-streams, and so on. Among these applications, image processing is relatively easier to implement and demonstrate. In this chapter, we implement an image processing system on FPGA and measure calculation outputs from the chip to calculate the actual processing error while reducing the energy. Processed images with reduced processing energy are compared with the image with no processing errors. This reveals that it is possible to apply this low-power design scheme while keeping the image quality acceptable.

The details of the design of the adder and multiplier with MSB-LSB weighted supply voltage scaling and eventually the whole MAC unit combining the two is presented in section 4.2. After an overview of the image processing approach used in the following demonstrations, several experimental trials to evaluate the effectiveness of inexact computing on FPGA hardware are presented in section 4.3. Measurement

results are given in section 4.4 with a final summary of the chapter in the last section.

4.2 ALU Design for MSB-LSB weighted supply voltage scaling

A 32-bit adder capable of MSB-LSB weighted scaling of supply voltage is shown in the previous chapter. This specific adder architecture demonstrates even better energy-efficiency compared to conventional adder architectures due to the following advantages. The adder architecture determines the outputs based solely on the input vectors of the current block; therefore, the calculation accuracy of the current block is less susceptible to calculation errors from the previous block compared to other adder architecture. Moreover, critical path delays in this adder architecture are paths for calculating the MSB blocks, so supply voltages for LSB blocks can be lowered relatively to those used in the MSB blocks even before considering any calculation error, in general. More importantly, in view of implementation perspective, our proposed scheme does not require that the critical-path blocks be separated from the non-critical-path blocks, thus adding minimal extra routing for each supply voltage. Since data-paths often are arranged in a bit-sliced organization, there is no additional floor planning necessary for this additional layout routing.

4.2.1 Adder Design

The eventual goal in this section is to design a building block for the digital signal processing of image information. A 32-bit MAC is chosen for this purpose; therefore,

our previous adder design is modified to process 64-bit wide data.

To extend the CCS scheme up to 64-bit data width, the 4-bit CCS adder is modified to an 8-bit version as shown in Figure 4.1. In the modified CCS architecture, the conditional carry outputs for each bit are selected by the multiplexers depending on the conditional carry signals of the two previous bits. As seen in Figure 4.1, in this configuration there are several back-to-back connections between the transmission-gate multiplexers. However, these multiplexers are not in the critical path and thus will not greatly affect the results of this study. Aside from this, the overall construction of the 64-bit adder is exactly the same as the 32-bit adder presented in the previous chapter. The only difference is the extension of the 4-bit CCS and CSS adders to the 8-bit version.

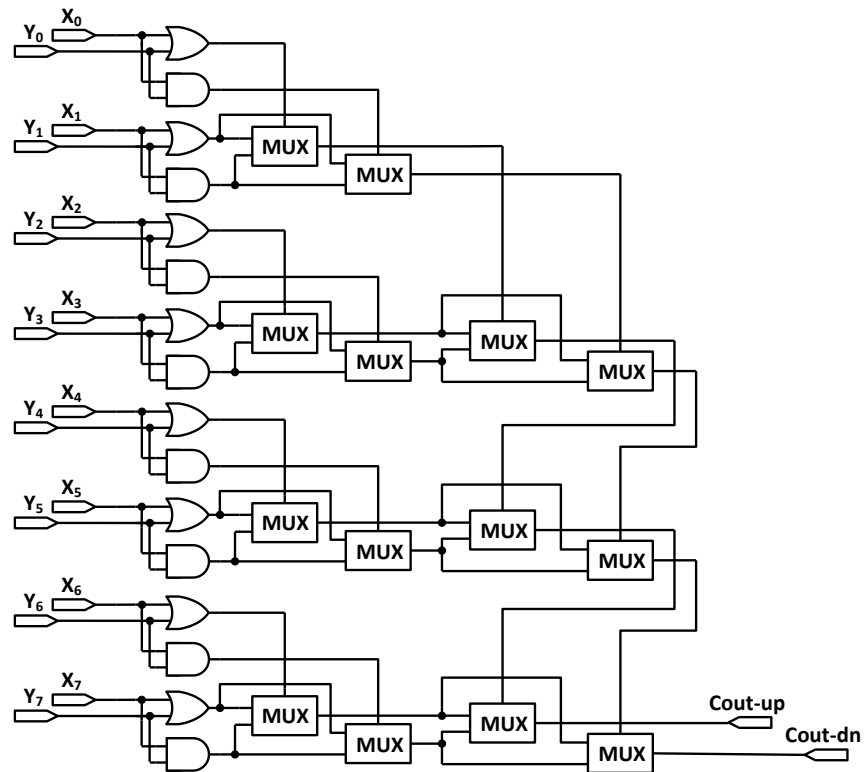


Figure 4.1 Logic implementation of 8-bit Conditional Carry Select adder block.

Figure 4.2 illustrates the relationship between energy and calculations error for the 32-bit adder case. The same simulation procedures used for the 32-bit adder are applied to the 64-bit adder. The 64-bit input vectors are generated and clocked at 3GHz. Again, the case of a single nominal supply voltage (1 V) is chosen as the baseline, i.e., free of computation errors. Various combinations of eight MSB-LSB supply voltage values are selected from the graph of Figure 3.15. The simulation results are plotted together with the results from the case of the 32-bit adder, as shown in Figure 4.2. As expected, the relationship is found to be similar to the 32-bit adder, with slight deviation expected due to the differences in circuit topology and slightly increased complexity of the circuit architecture.

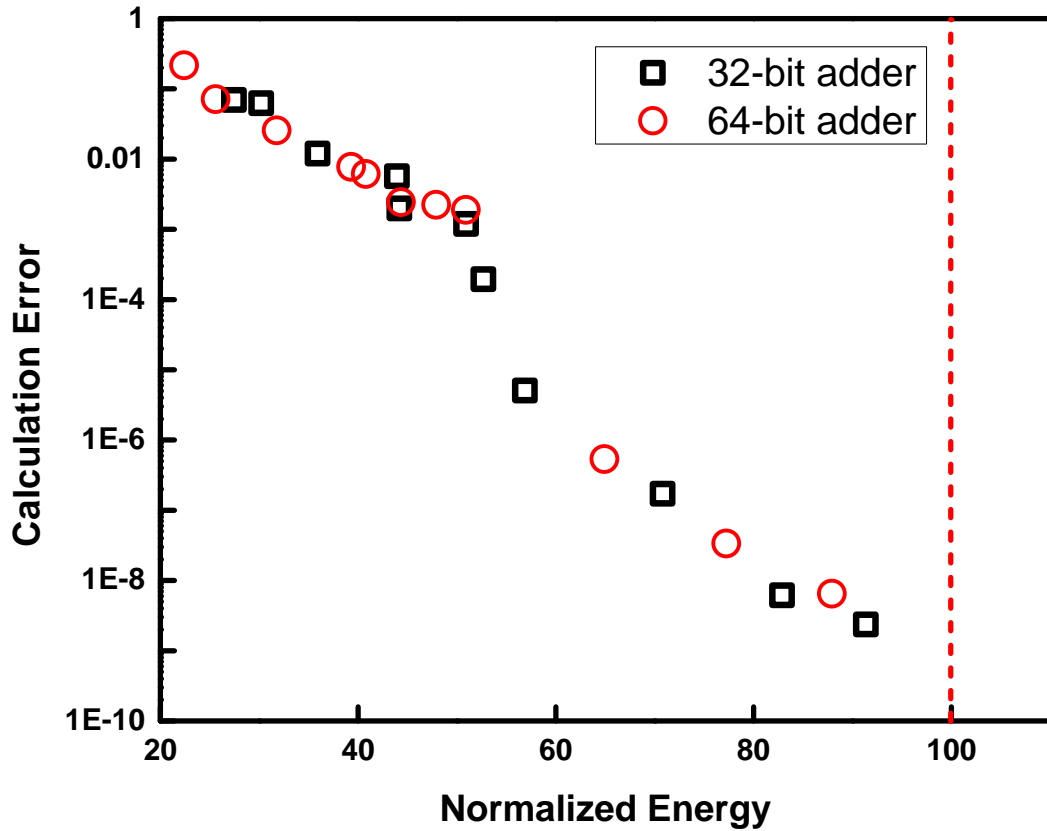


Figure 4.2 Energy vs. calculation error for 64-bit adder by applying MSB-LSB weighted scheme. A 32-bit adder result is plotted together for comparison.

4.2.2 Multiplier Design

Multiplication is less common than addition, but it is still essential for microprocessors and even more critical for digital signal processors and graphics engines, where most of the computing power is used for multi-dimensional matrix manipulation. The most basic form of multiplication consists of forming the product of two unsigned binary numbers and summing the partial products by column as shown in Figure 4.3.

$$\begin{array}{r}
 \begin{array}{r}
 \text{Multiplicand} \\
 1\ 0\ 1\ 0\ 1\ 0 \\
 \times \quad \text{Multiplier} \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 1\ 0\ 1\ 0 \\
 1\ 0\ 1\ 0\ 1\ 0 \\
 0\ 0\ 0\ 0\ 0\ 0 \\
 +\ 1\ 0\ 1\ 0\ 1\ 0 \\
 \hline
 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 0
 \end{array}
 \end{array}
 \begin{array}{l}
 \\
 \\
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{Partial products} \\
 \\
 \text{Result}
 \end{array}$$

Figure 4.3 Multiplication operation example.

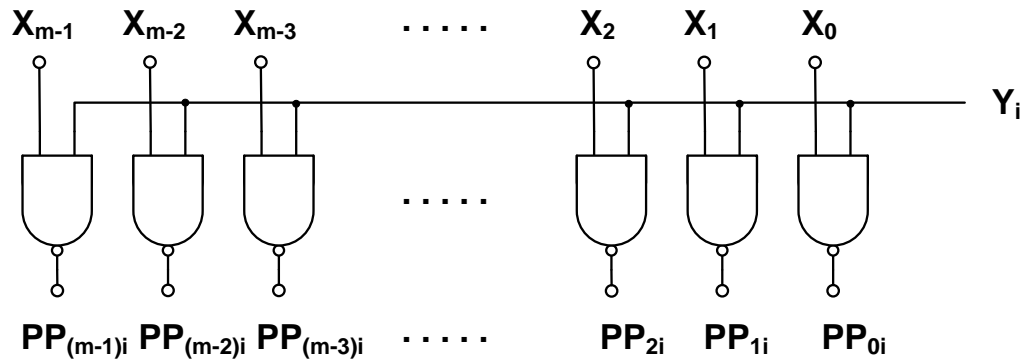


Figure 4.4 Partial product generation logic. $M \times N$ -bit multiplication produces $M \times N$ numbers of AND (or (NAND) for this partial product generation. Since Radix 2^r multipliers produce $M \times N/r$ numbers of partial products, fewer partial products lead to a smaller and faster CSA array.

A circuit implementation of partial products generation can be done relatively easily by employing a series of 2-input AND gates as in Figure 4.4. However, summing partial products is slow without employing more efficient parallel approaches. Therefore, in this section, Radix-4 Booth encoding [65] and a 4:2 compressor are employed to boost the speed of the partial products summation.

Radix 2^r multipliers [66] produce N/r partial products, each of which depends on r bits of the multiplier. Fewer partial products lead to a smaller and faster carry save adder (CSA) [67]. Table 4.1 shows how the partial products are selected based on different combinations of the input bits of the multiplier.

Inputs			Partial Product	Booth Selects		
X_{2i+1}	X_{2i}	X_{2i-1}	PP_i	$SINGLE_i$	$DOUBLE_i$	NEG_i
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	2Y	0	1	0
1	0	0	-2Y	0	1	1
1	0	1	-Y	1	0	1
1	1	0	-Y	1	0	1
1	1	1	-0 (= 0)	0	0	1

$$SINGLE_i = X_{2i} \oplus X_{2i-1}$$

$$DOUBLE_i = \sim(X_{2i+1}) \cdot X_{2i} \cdot X_{2i-1} + X_{2i+1} \cdot \sim X_{2i} \cdot \sim X_{2i-1}$$

$$NEG_i = X_{2i+1}$$

Table 4.1 Radix-4 modified Booth encoding values and the corresponding Boolean expressions

For $SINGLE_i$, a 2-input XOR gate is the obvious choice for circuit implementation and NEG_i is just an inverter. For generation of $DOUBLE_i$ signal, there are two possible implementation styles: one is a cascade of a 3-input NAND followed by a 2-input NAND, and the other one is a cascade of a 2-input NAND and a transmission-gate multiplexer. Both are functionally equivalent, however, some different simulation results are acquired in terms of performance, power, and area as summarized in Table 4.2. IBM's 45-nm FDSOI process technology is employed for this comparison. The cascade of a 2-input NAND and MUX is selected for the 32-bit MSB-LSB weighted supply voltage scaling multiplier in this section since its implementation area is smaller.

	3NAND-2NAND	2NAND-MUX
Average Delay (ps)	8.9	9.5
Max Delay (ps)	9.6	10.2
Min Delay (ps)	8.7	7.8
Standard deviation (ps)	0.465	1.139
Power (μ W)	22.72	23.24
Area (# of unit sized Tr)	38	25

Table 4.2 Simulation results for both circuit implementations. Performance and power are comparable for both cases; however, area is much smaller when using the 2NAND-MUX implementation.

	2NAND-2NAND	AOI22-INV	2NAND-PTL	2NAND-TRANS
Average Delay (ps)	7.59	7.56	12.10	12.13
Max Delay (ps)	9.11	11.9	15.1	18.9
Min Delay (ps)	6.67	5.34	8.40	8.24
Std. deviation (ps)	0.753	1.887	2.513	3.919
Power (μ W)	24.024	30.316	29.163	32.812
Area (# of Tr)	24	27	24	28

Table 4.3 Simulation results for various circuit implementation styles for Booth selector.

The Booth selector can also be implemented in a variety of ways and the results of a similar comparison for finding the optimal implementation is summarized in Table 4.3. These results indicate that two back-to-back 2-input NAND gates are optimal for implementing the Booth selector due to its superior performance and power consumption when compared to the other implementations.

The Booth encoder and selector are optimized and used in the 32-bit multiplier as a building block. During this optimization, the highest priority was to minimize implementation area since these basic building blocks are instantiated in large numbers and could impact the layout area significantly. The final circuit schematics are shown in Figure 4.5 and Figure 4.6.

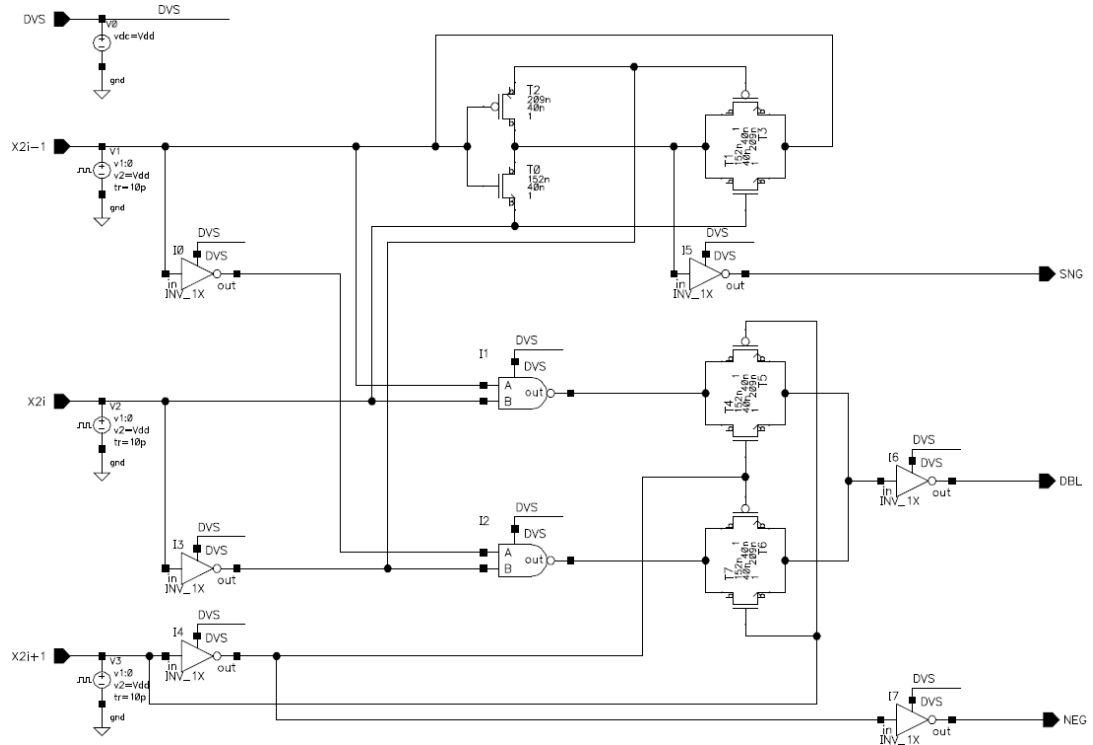


Figure 4.5 An area-optimized circuit implementation for modified Radix-4 Booth encoder. The 2-input XOR gate is also implemented using transmission-gates due to its compact layout.

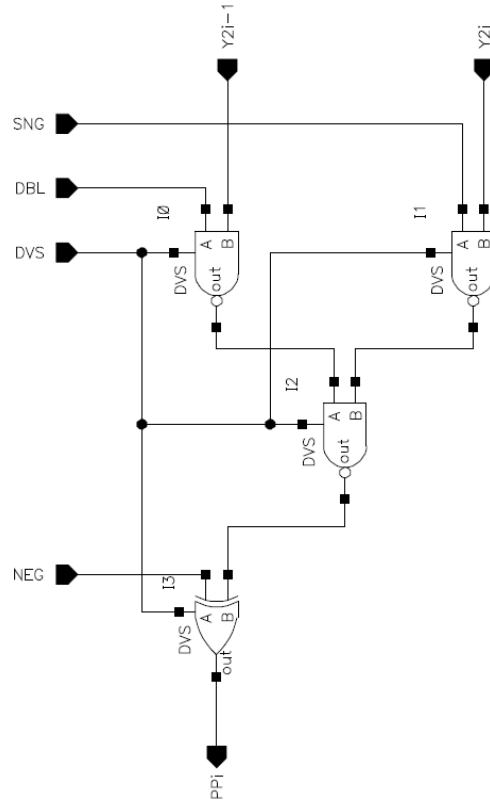


Figure 4.6 A schematic of circuit implementation for modified Radix-4 Booth selector.

Also, in this 32-bit multiplier, 4:2 compressors are used. Such compressors are preferred in a binary tree to produce a more regular layout [68]. A 4:2 compressor takes four inputs of equal weight and produces two outputs. Although it generates an intermediate carry, t_i , into the next column and accepts a carry, t_{i-1} , from the previous column, this horizontal path does not directly impact the delay because the output of the top CSA in one column is the input of the bottom CSA in the next column. The symbolic representation of the 4:2 compressor emphasizes only the primary inputs and outputs to focus on the main function of reducing four inputs to two outputs. Based on the number of bits N , the number of 4:2 compressors required is given by

$$\left\lceil \log_2 \left(\frac{N}{2} \right) \right\rceil.$$

Despite the significantly increased critical path delay that results from using a 4:2 compressor compared to a CSA, this is offset by the significantly fewer number of required levels when using this implementation compared to traditional CSA's. In this 32-bit multiplier design, only four levels of 4:2 compressor are required. The regular layout and routing also make the binary tree attractive. Figure 4.7 shows the schematic view of the transmission-gate implementation of a 4:2 compressor [69]. It uses only 48 transistors, allowing for a smaller multiplier array with shorter wires. Due to its layout compactness, this type of 4:2 compressor is used entirely for the 32-bit MSB-LSB weighted scaling multiplier.

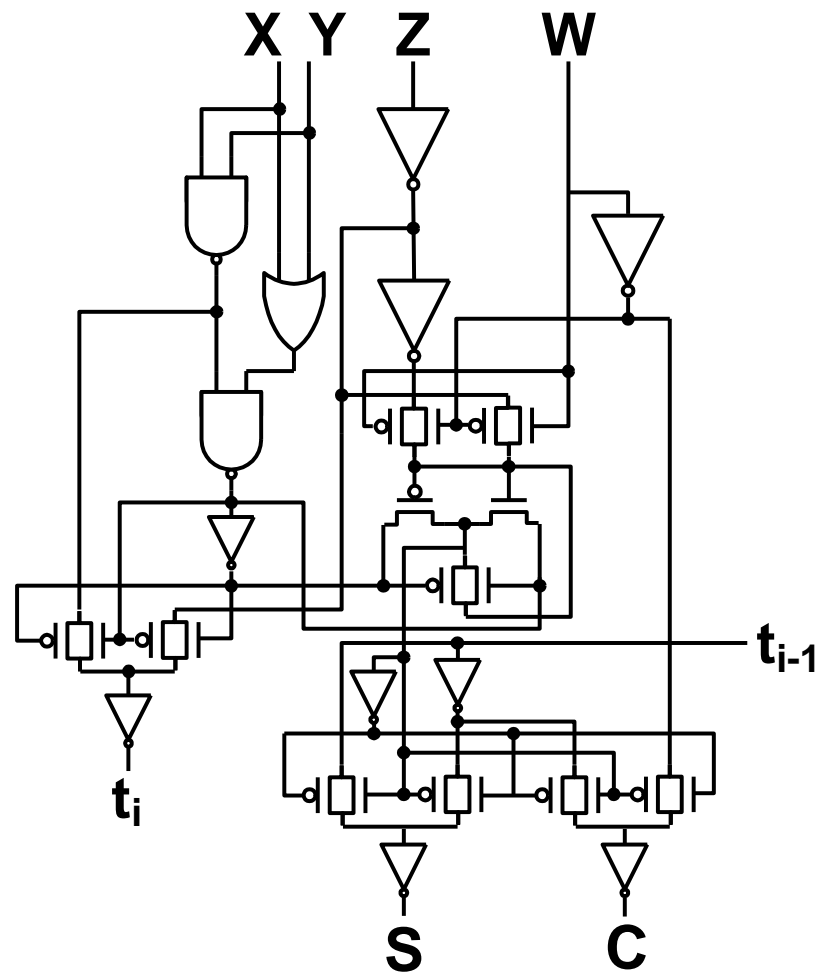


Figure 4.7 Transmission-gates 4:2 compressor. Note that it uses three distinct XNOR circuit forms and two transmission-gates multiplexers.

In summary, the implementation highlights for the 32-bit multiplier are as follows.

- (a) Radix-4 Modified Booth Encoder/Selector
- (b) 3 cascades of 4:2 compressors
- (c) 3:2 compressor for redundant output
- (d) 64-bit addition at the final stage

To have a similar reference for comparison, the 32-bit multiplier is divided into two functional sub-blocks: the first sub-block includes the partial product generation and the partial product sum as one functional block and the second sub-block consists of the final addition block. Since the critical path delays for both of divided blocks are comparable, these divided blocks are connected as a pipeline and similarly clocked at 3GHz. The MSB-LSB scheme is then applied to each block separately. First, an MSB-LSB weighted scheme is employed when calculating the column sum as shown in Figure 4.8. This figure illustrates the case of 16x16 multiplication operations for simplicity, however, actual implementation is 32x32 multiplication and each MSB-LSB weighted block is divided into 8 sections, which operates a total of 8 column sums. Second, a 64-bit MSB-LSB weighted scaling adder, which was presented in section 4.2.1, is employed for final addition.

Again, to evaluate the calculation errors, the same simulation framework is applied to this 32-bit MSB-LSB multiplier as performed for the 32-bit and 64-bit adders in the previous chapter. Simulation results are presented in Figure 4.8. Since the weighted supply scheme is applied twice per functional arithmetic operation, the gains in energy-efficiency is even greater than the gains observed in the 32-bit adder in the

previous chapter, but the overall calculation error also increases comparably due to the multiplicative cascading of calculation errors.

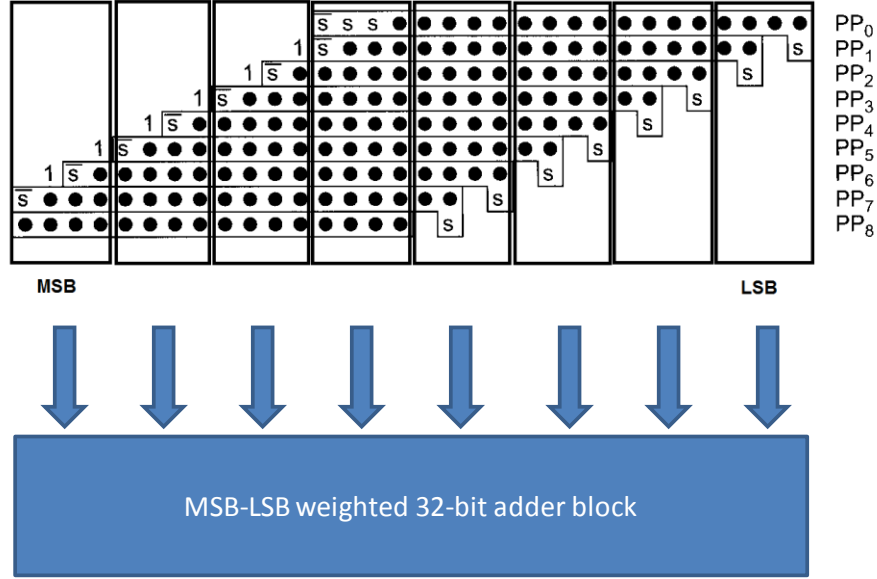


Figure 4.8 Application of MSB-LSB weighted scaling scheme at the stage of column sum for the case of 16-bit multiplier

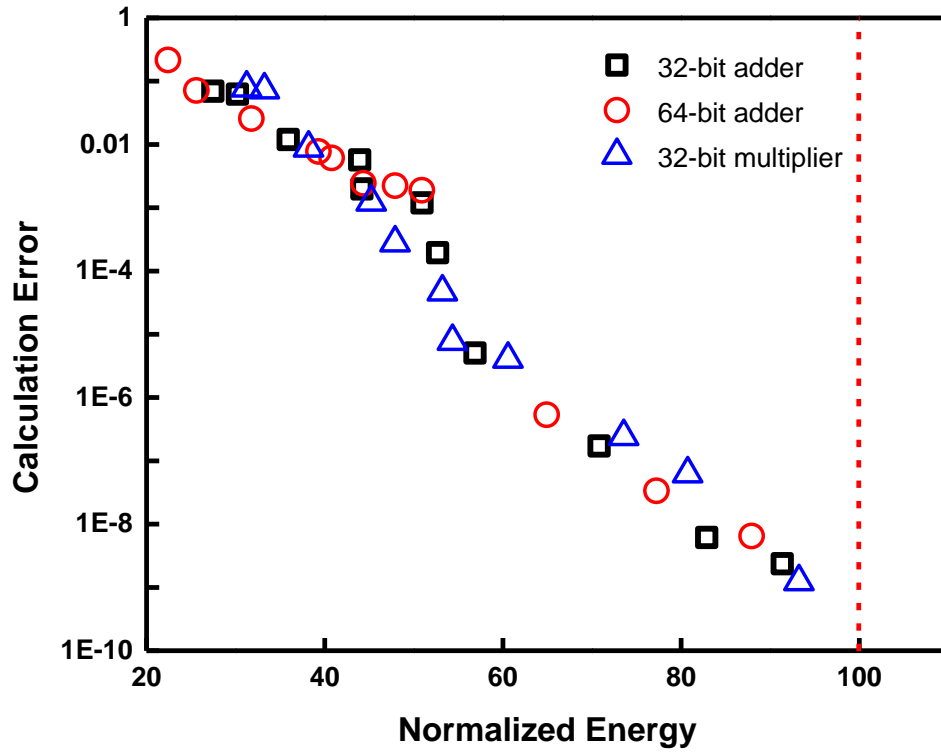


Figure 4.9 Energy vs. calculation error for the 32-bit multiplier by applying MSB-LSB weighted scheme as shown in Figure 4.8.

4.2.3 Multiplier-Accumulator for Digital Signal Processing

Now that the design details of the 32-bit multiplier and 64-bit adder are ready, a 32-bit Multiplier-Accumulator (MAC) using these two basic arithmetic building blocks can now be implemented. This can then be used as a vehicle to demonstrate the enhanced energy-efficiency using the proposed inexact computing methodology for digital signal processor. Employing this 32-bit MAC equipped with the MSB-LSB weighted scaling scheme, our proposed ultra-low power design methodology can be further demonstrated in a real world image processing application since MAC's are also used in such applications as a fundamental component of the FIR filter [70]. This will be explained further in the next section. As shown in Figure 4.10, the MAC is implemented by instantiating one 32-bit multiplier, one 64-bit adder, and 64-bit wide registers.

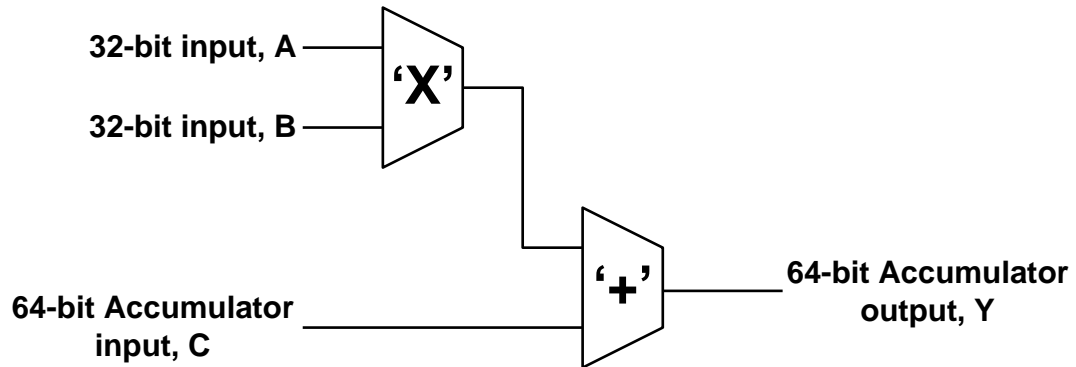


Figure 4.10 MAC implementation for $Y = A \cdot B + C$.

At this point, it is worth noting that fixed-point and floating-point arithmetic is not supported in this ALU design as it is currently implemented. This is because 32-bit double-precision numbers are implemented using two 32-bit memory locations in

modern 32-bit computers [71]. The proposed ALU design is not currently supporting these features, so will not be employed for digital signal processing of images in this work. This will be discussed again in the next chapter.

4.3 Image Processing Example: Inexact Computing

Digital image processing has many practical applications ranging from high-end image systems mounted on spacecraft or used in medical and clinical applications to low-end consumer electronics such as TV's, video conferencing, cameras, and so on. Especially in handheld electronics, digital image processing holds one of the most important positions in handling our daily computing. Examples include many features that are common in mobile devices, including video conferencing via wireless communication networks, addition of special effects to digital pictures taken by embedded cameras, image compression for storage, streaming or decompression for viewing and playing, software applications for pattern recognition, and many others. Computing demand for increased image bandwidth and faster processing imposes a significant on the minimum achievable power consumption.

There are numerous applications where a low error rate can be tolerated. One particular example is human vision, as a certain level of information is sufficient to saturate the perception of the human brain [46]. This observation is a major motivation to choose image signal processing as an experimental framework to demonstrate the advantages of our proposed ultra-low power methodology in this chapter.

4.3.1 Experiment and Measurement Scheme

The overall goal of this experiment is to demonstrate the inexact computing concept by comparing the image signals processed using actual hardware at different supply voltages. By doing this, we can determine the maximum calculation error threshold for the human vision system above which users will notice quality degradation of the processed images. It can then be possible to reduce the power consumption to a level where this maximum error rate is supported, i.e., when the human vision system begins detecting the unacceptable errors on the processed images.

Hard macro DSP engines exist in general-purpose CPU's such as the AVX instruction set in Intel's Core i7 products [72], in special-purpose microprocessors like Texas Instrument's C6000 DSP products [73], and even in field-programmable gate arrays (FPGAs). For this experiment, Xilinx's Virtex-6 FPGA chip [74] is chosen because of its programming flexibility and re-configurability. Historically, FPGAs have been slower, less energy efficient and generally achieved less functionality than their fixed ASIC counterparts. A study by Rose [75] had shown that designs implemented on FPGAs need on average 40 times more area, draw 12 times more dynamic power, and are 3 times slower than the corresponding ASIC implementations. However, today's state-of-the art FPGAs fabricated using 28-nm process technology are narrowing this gap between FPGAs and ASIC solutions of older generations by providing significantly reduced power, increased speed, lower BOM cost, minimal implementation real-estate, and maximum on-the-fly configurability. At the time of this study, the most recent FPGA on the market is Virtex-6, which is manufactured

using TSMC's 40-nm bulk CMOS technology [76]. Some features of Virtex-6 such as advanced DSP48E1 slices including 25×18 , two's-complement MAC units with optional pipelining and dedicated cascade connections [77] and programmable dual/single-ported embedded SRAM [78] are particularly attractive for this experiment.

We perform all experiments on Xilinx's Virtex-6 FPGA Embedded Kit [79] and ISE Design Suite: System Edition [80]. The raw image file for image processing is a 640×480 , RGB color image in JPEG format. Edge detection and enhanced sharpness are used for evaluating the calculation errors. Inverse gamma correction is applied to the raw image to restore its linearity color space prior to applying the two image processing algorithms. Gamma correction is also applied to the processed images. In this experiment, a gamma correction coefficient of 2.2 is used.

In digital image processing, edge detection and sharpness enhancement operations can be achieved through the process of spatial domain filtering [81]. Spatial domain filtering simply indicates that filtering process takes place directly on the actual pixel of the image itself. Filters act on an image to change the values of the pixels in some specified way and are generally classified into two types: linear and nonlinear. A linear filter is more common and is the only focus of this work. This is also why the non-linear gamma corrected images are converted back to linear images using inverse gamma correction. It is always recommended to make the corresponding images linear before applying a linear filter. Whenever applying spatial domain filtering to the images, the value of the target pixel is then replaced by a new value which depends only on the value of the pixels in a specified neighborhood around the target pixel.

An important measure in images is the concept of connectivity. Many operations in image processing use the concept of a local image neighborhood to define a local area of influence, relevance, or interest. Central to this theme of defining the local neighborhood is the notion of pixel connectivity, i.e. deciding which pixels are connected to each other. When we speak of 4-connectivity, only pixels which are N, W, E, S of the given pixel are connected. For cases where the pixels on the diagonals must also be considered, we then have 8-connectivity (i.e. N, NW, W, NE, SE, E, SW, S are all connected). Operations performed locally in images, such as image sharpening and edge detection, all consider a given pixel location (i, j) in terms of its local pixel neighborhood indexed as an offset $(i \pm k, j \pm k)$. The majority of image processing techniques currently use 8-connectivity by default, which for a reasonable neighborhood size is often achievable in real time on modern processors for the majority of operations. Filtering operations over a whole image are generally performed as a series of local neighborhood operations using a sliding-window-based principle, i.e. each and every pixel in the image is processed based on an operation performed on its local $N \times N$ pixel neighborhood (region of influence).

In linear spatial filters the new or filtered value of the target pixel is determined as some linear combination of the pixel values in its neighborhood. The specific linear combination of the neighboring pixels that is taken is determined by the filter kernel (often called a mask). This is just an array/sub-image of exactly the same size as the neighborhood containing the weights that are to be assigned to each of the corresponding pixels in the neighborhood of the target pixel. Filtering proceeds by successively positioning the kernel so that the location of its center pixel coincides

with the location of each target pixel, each time the filtered value being calculated by the chosen weighted combination of the neighborhood pixels. This filtering procedure can thus be visualized as sliding the kernel over all locations of interest in the original image (i, j), multiplying the pixels underneath the kernel by the corresponding weights w, calculating the new values by summing the total and copying them to the same locations in a new (filtered) image as demonstrated in Figure 4.8. The mechanics of linear spatial filtering actually express in discrete form, a process called convolution. For this reason, many filter kernels are sometimes described as convolution kernels, it then being implicitly understood that they are applied to the image in the linear fashion described above.

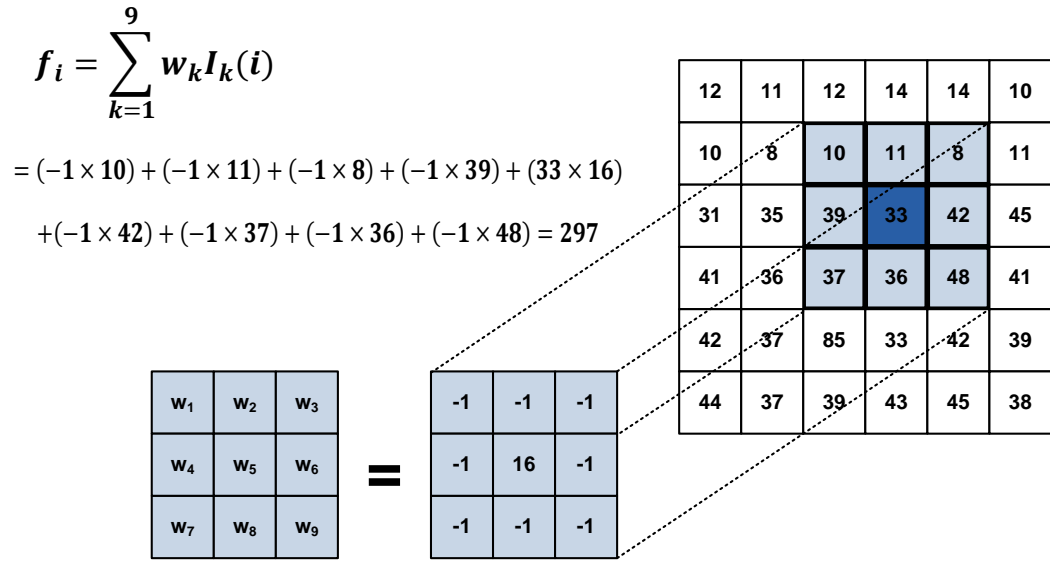


Figure 4.11 The mechanics of image filtering with an $N \times N = 3 \times 3$ filter. In this specific case, the original target pixel value of 33 is filtered to an output of 297. Usually, there will be a coefficient to be multiplied by the filter, for example, $1/16$ for sharpness enhancement. The filtered value will not be greater than 255, which is the maximum number can be represented with 8-bit RGB color image.

Figure 4.12 shows an actual image filter used for implementation of sharpness enhancement filter and shows the original image and filtered image together.

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \frac{1}{32} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 16 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

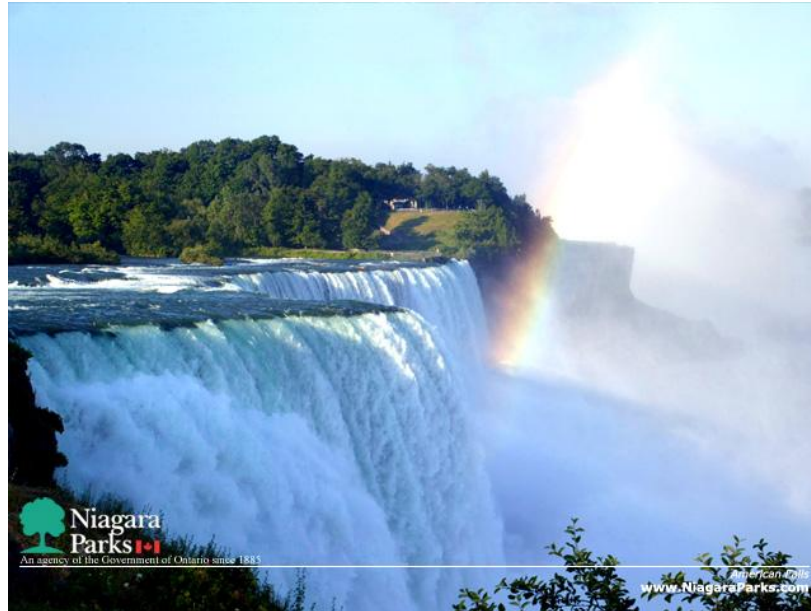


Figure 4.12 3 x 3 image filter for sharpness enhancement. Original image (upper) and processed image (lower) after applying the above 3 x 3 image filter. Courtesy of New York State Parks

4.3.2 Experiment Using Manufacturer-supplied Design Platform

The first trial for this experiment is executed using the pre-compiled source and software framework provided by Xilinx. Although it is tuned and modified for the specific goal of this experiment, if successful, it would be the best way since it requires the least time and effort to perform this experiment.

The Base Reference Design [82] provided by Xilinx is made to filter images that are transferred via Ethernet between the evaluation board and a PC. The images are stored in DDR3 SDRAM on the board. The stored image is continuously read from SDRAM, filtered by the FPGA, and the resulting image is continuously stored back in the DDR3 SDRAM. This filtered image is then retrieved by the Base Reference Design Interface Software and displayed on a PC.

Figure 4.13 shows a block diagram of the base reference design that has been implemented in the Virtex-6 FPGA. The reference design includes common functions for Ethernet communication, external memory interface, UART, and control. A DDR3 Memory Controller Block is used to store both the unfiltered and filtered images in the DDR3 SDRAM. These images are sent from a PC via a series of Ethernet packets. This memory controller is continuously reading, filtering, and storing images back into this memory. The PC also periodically retrieves the filtered images via Ethernet for display. The Ethernet Management section includes an on-chip hard coded MAC and a Packet Processing Engine. This provides a way to control various aspects of task processing such as transferring images between the board and a PC, and receiving the status from the board. A simple MDIO (Management Data Input/Output) controller is

implemented using a Xilinx PicoBlaze™ processor [83]. The purpose of this controller is to determine the presence of an Ethernet link as well as its operating speed. The image processing structure consists of a 5x5 pixel 2D FIR filter.

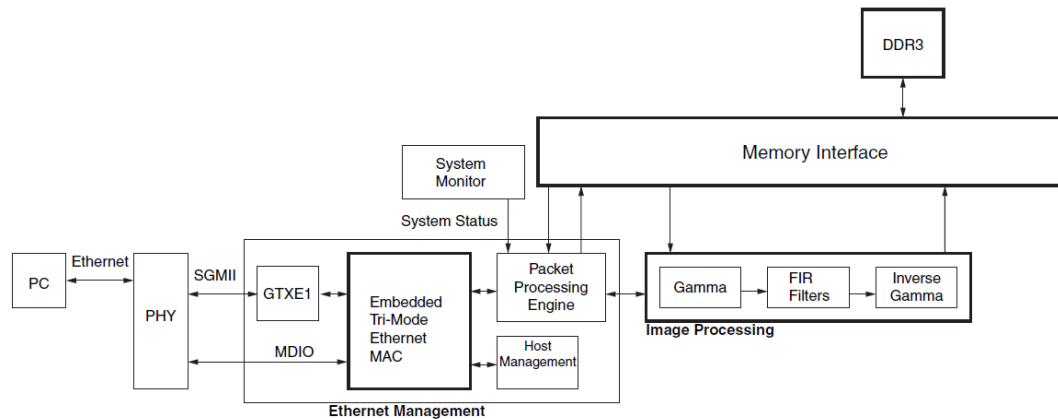


Figure 4.13 Base Reference Design Block Diagrams [82].

We cannot exploit this Base Reference Design directly because we require the capability of varying the voltages supplied to FPGA. To adjust the value of supply voltage, modification of Verilog HDL and design are needed. First, to accommodate the capability of varying V_{dd} , Texas Instrument's TUSB3210 module [84] is connected to the board through the JTAG port and the company's Fusion software [85] is employed to control the supplied voltage values and monitor power and temperature. Second, an internal routine to hold this task prior to processing image signals in the DSP block to control the supply voltages must be implemented. Since the functionality of the other blocks is not guaranteed with the lowered supply voltages, the application of lowered voltage values is confined to the block of interest. This scheme is explained in Figure 4.14. Additional BlockRAM is placed in between the DDR3 SDRAM and DSP slices to store the data temporarily.

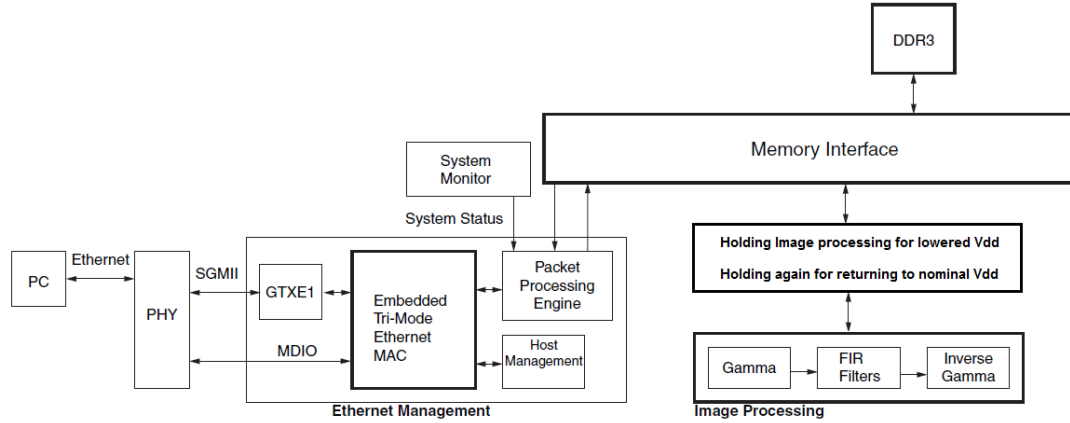


Figure 4.14 Modification to the Base Reference Design Block to accommodate the desired experiment.

The following is a summary of the operating principles for this experiment. First, hold the pipelined DSP data-path before image processing and then notify to user that it has stopped. The system then waits for user input. After adjusting the value of V_{dd} , the user can push a button to resume image processing. At this time, the image processing is performed with lowered supply voltage and the processed image is saved again on the newly instantiated SRAM block, which is located between the image processing block and the DDR3 SDRAM. Once saving is done the system is held again and notifies the user of completion of image processing. Then, the user adjusts V_{dd} value back to its nominal value to guarantee the proper functionality of other blocks. The user pushes the button again to continue processing for the rest of the board. The processed and saved image is sent to DDR3 and to PC via Ethernet.

Changes in software are required to implement the above idea; however, the manufacturer doesn't supply the source code for handling the image data via Ethernet. It is therefore not possible to accommodate our measurement approach within this platform. Due to this difficulty, the first approach is discarded and switched to using the UART interface rather than Ethernet since MATLAB embeds the capability of

programmable serial interface on the host PC [86].

Figure 4.15 shows the basic functional diagram for this implementation option. Additionally, simplified functional blocks are expected to reduce the dependency on other peripherals such as Ethernet, DDR memory controller, DDR SDRAM, and temporal storage. More details will be discussed in the next section.

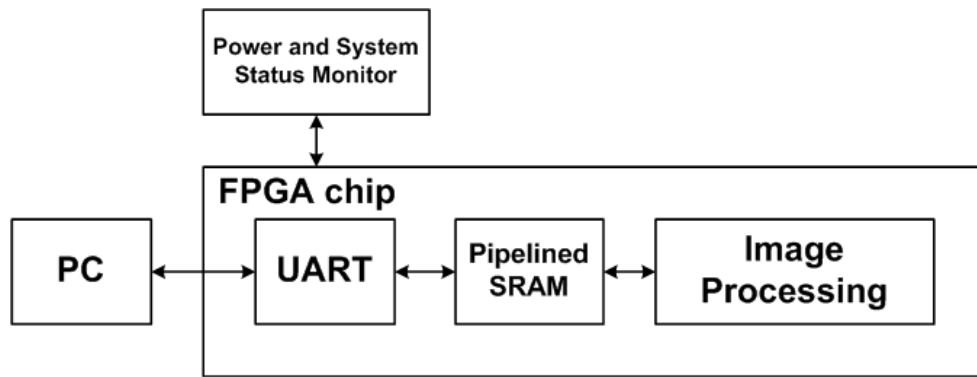


Figure 4.15 As another modification to get rid of Ethernet software dependency, the UART interface to PC is selected.

4.3.3 Experiment using Conventional FPGA Design Flow

The first attempt planned in the previous section proved to be a failure since no customer support from the manufacturer is available. Even if there were support from Xilinx, it is still unlikely that successful measurement would have been achievable. The reason for this conclusion will be clarified in this section after attempting the measurement again with an even simpler FPGA implementation.

The previous section ends with a suggestion for the next experiment approach. The rough idea of Figure 4.15 is implemented using the conventional design flow of Virtex-6 FPGA while relying on software tools such as ISE Design Suite, Xilinx

Platform Studio [87], Xilinx Software Development Kit [88], and System Generator for DSP [89]. Compared to the manual tweaking of pre-developed HDL code in the previous section, the required design is implemented by writing the full HDL codes from the beginning.

Prior to proceeding with the second approach, prior knowledge of the effective supply voltage ranges for proper functionality of the bigger SRAM block must be known since in this implementation style, the role of the large block RAM is critical. For this purpose, we first implement the test platform described as in Figure 4.16.

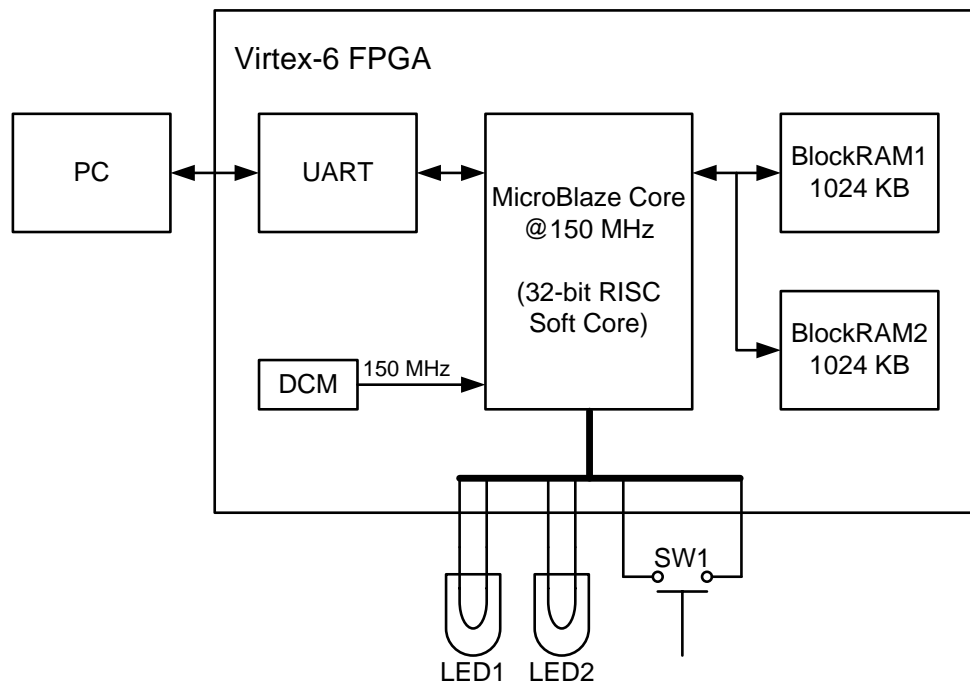


Figure 4.16 Test platform implemented using Virtex-6. Two 1MB of large block RAM, 32-bit MicroBlaze core operating at frequency of 150MHz, Digital Clock Manager, and UART are implemented by HDL codes generated by Xilinx’s ISE Design platform.

In this implementation, UART and MicroBlaze [90] are instantiated as soft IP, which means generic logic and memory elements are synthesized, placed, and routed

for functionality by software tools. DCM (Digital Clock Manager) [91] and BlockRAM [92] are, on the other hand, Hard IP, which means the functional block was already in the FPGA chip as pre-fabricated hardware and routed externally only by software tools.

To find the V_{dd} ranges for correct memory access, the original color image is transferred via UART and saved in BlockRAM 1 at nominal V_{dd} of 1 V. LED 1 is on and the task is halted simultaneously when the file transfer is completed. V_{dd} is lowered and SW 1 is pushed, and then the contents of BlockRAM 1 are copied into BlockRAM 2 at lowered V_{dd} . LED 2 is on when the copy process from BlockRAM 1 to BlockRAM 2 is finished, and the task is halted again. After turning the supply voltage back to the nominal value, push of SW 1 makes the system send the contents of BlockRAM 2 to PC via UART. Comparison between the original file and the received file from the BlockRAM 2 confirms the correct operation of BlockRAM control at the lowered supply voltage. MicroBlaze core is synthesized at the operating frequency of 150MHz. Clock frequencies to the other blocks should be adjusted to this lowest clock frequency although BlockRAM can ideally operate up to 600MHz. The results of this experiment show that the operational V_{dd} range is from 0.84 V to 1.1 V.

With this V_{dd} range for guaranteed BlockRAM operation, we design the image processing system clocked at 150MHz. A detailed block diagram is displayed in Figure 4.17. A 2-dimensional 5 tap FIR filter block for image sharpness is re-used here without any modification from the first experiment.

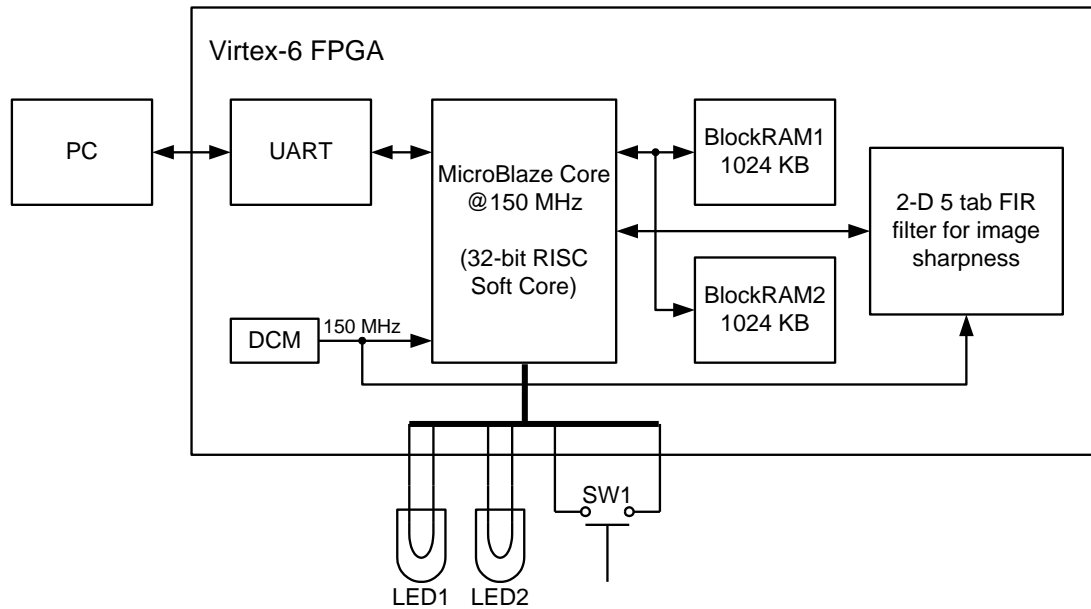


Figure 4.17 Image processing block implemented at 150MHz. Image processing block is added to the block of Figure 4.16.

The experiment procedure now for investigating the effect of supply voltage scaling is exactly the same as the previous experiment for finding the V_{dd} ranges of correct SRAM operation. The saved image on BlockRAM 1 is loaded into the FIR filter and processed by the control signal generated by the MicroBlaze core operating at 150 MHz and the lowered V_{dd} . Again, processed data is saved on BlockRAM 2 and V_{dd} is reset to its nominal value prior to sending the contents of BlockRAM 2 back to PC via UART interface.

The raw data from this experiment is processed using MATLAB and plotted as shown in Figure 4.18. There is no calculation error with the supply voltage values from 0.84 V to 1 V, if the FPGA is operating at 150MHz. This result indicates that the image processing block, which consists of 2-D 5 tap FIR filters, is not affected from a lowered supply voltage down to 0.84 V. According to Xilinx's specification sheet,

DSP48E1 slices, which are core components for FIR filter implementation, can also operate up to 600MHz ideally [93]. No processing errors at 150MHz seems to be plausible due to the DSP Hard MACRO's native performance. This shows there is still room to push the operating speed of the image processing block.

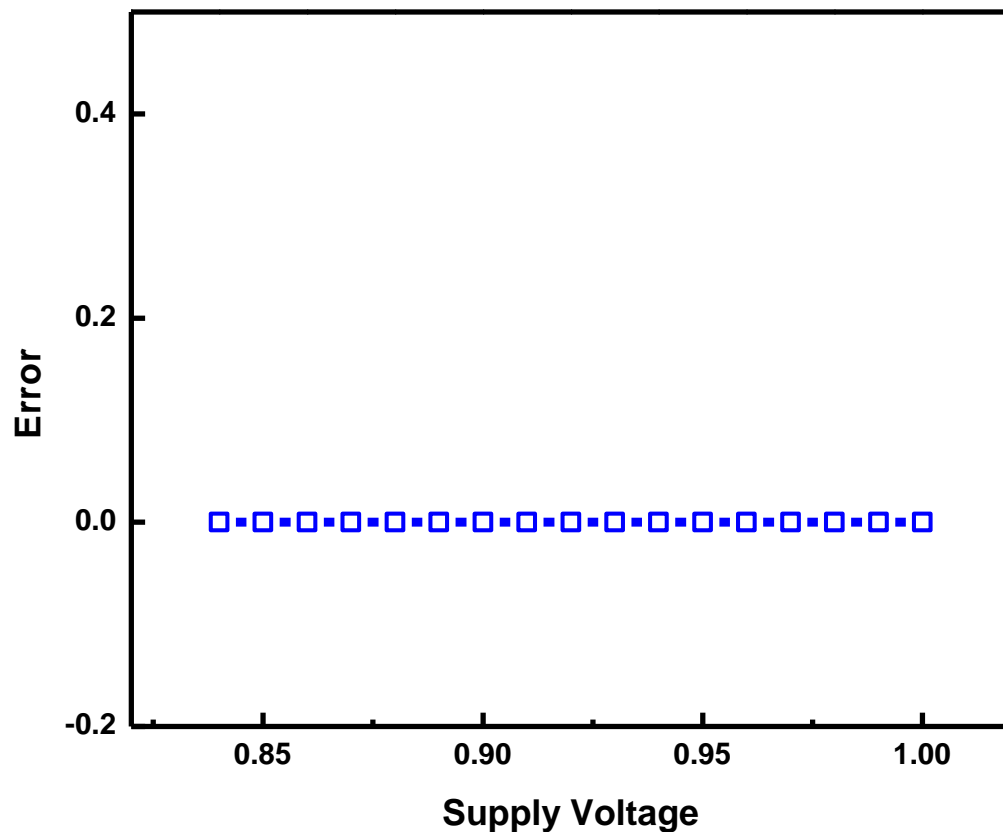


Figure 4.18 Supply voltages vs. image processing error. No calculation error in the range of V_{dd} range of 0.84 to 1 V.

In this experiment, the operating frequency of the MicroBlaze core is bottleneck limiting the overall performance of image processing block. However, the reason why the core clocked at 150MHz previously is merely a problem of software. In the current version of ISE Design Suite and the evaluation board, 150MHz is the highest possible operating frequency for the soft IP version of 32-bit MicroBlaze RISC CPU core.

Another scheme to increase the operating frequency of the image processing block

is suggested as in Figure 4.19. In this scheme, the clock of the MicroBlaze core is fixed at 100MHz, which is decreased due to the increased synthesis complexity, but the clock to the memory and DSP blocks are raised up to 250MHz.

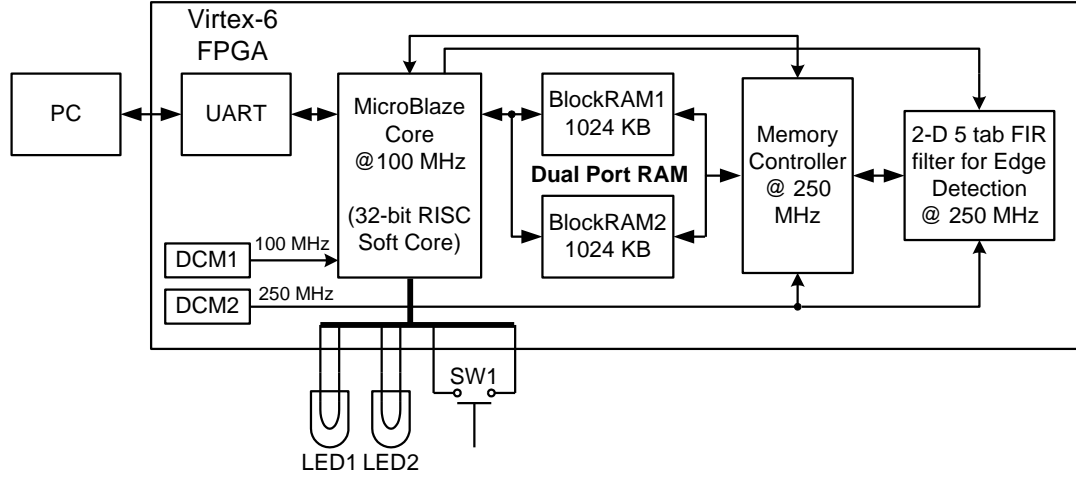


Figure 4.19 Modified scheme for increasing the operating frequency of the BlockRAMs and image processing block. Two different DCMs are used for 100MHz and 250MHz clocks, respectively.

In this separate clock scheme, frequencies of 100MHz for peripheral circuits and 250MHz for the DSP blocks are used. For image processing, image data is loaded and stored by an added memory controller operating at 250MHz. The clock frequency for the DSP48E1 slices is estimated by ISE Design Suite simulator to be about 250MHz.

The results from this experiment are plotted in Figure 4.20. Even with varying V_{dd} from 0.84 to 1 V, there is only a specific frequency value for determining the correct operation of the FPGA chip. We repeated the experiment with the same set-up 1,000 times with the same results. Failure to show the expected relationship between frequency (or supply voltages, power consumption) vs. calculation error is believed to result from the software dependency of the conventional FPGA design flow. Another failure to produce the relationship between energy vs. errors leads to a new suggestion

of experiment based on the scheme in Figure 4.21.

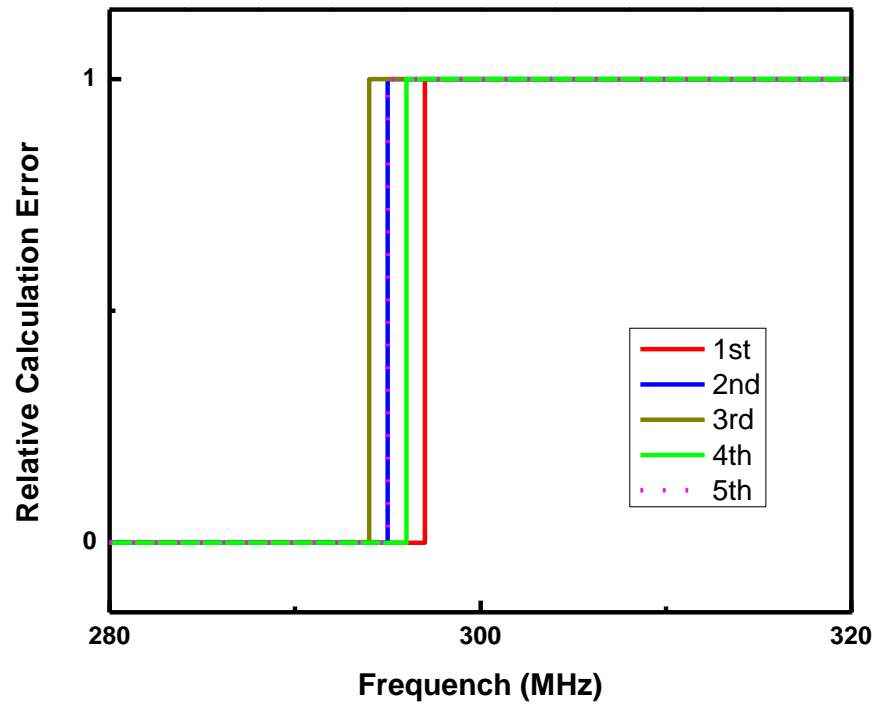


Figure 4.20 Frequency vs. error rate for the increased operating clock frequency of 250MHz.

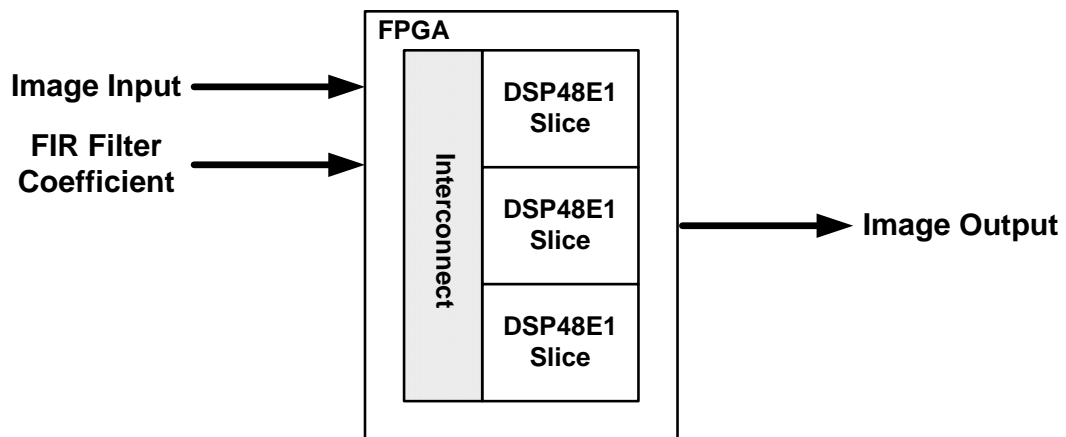


Figure 4.21 A new proposed method to minimize the software dependency of the FPGA implementation flow.

In this approach, we employ only the Hard IP part of the FPGA chip. By doing this, we can minimize the involvement of synthesis, place, and route design dependency imposed by a general FPGA design flow. Each 24-bit (8-bit RGB) image input pixel can be processed along with FIR filter coefficients to produce a desired image output. Without the software dependency of the FPGA, all input data should be applied via external input pins and probed at output pins for error calculation.

4.3.4 Experiment using Minimal Hardware Implementation

As suggested at the end of previous section, we explore a different option for implementing the experiment platform. A detailed set-up is illustrated in Figure 4.22. The minimal hardware requirement for FPGA is employed and MATLAB is used to process the raw data. The original JPEG file is fed to MATLAB and converted to RGB format. Prior to applying the linear image filter, inverse gamma correction is performed as described in the previous sections. For this calculation, integer RGB data is converted again to fixed-point data. Since the DSP48E1 slices in the Virtex-6 supports 18-bit fixed-point calculation [93], the digits are formatted to support 18-bit arithmetic manipulation. This bit-manipulation is extracted from MATLAB for generating the corresponding input pulse trains, which are directly applied to the input pins of the FPGA. The Moving Pixel Company's PG3A [94], a programmable pulse generator capability of up to 300MHz for 64 channels, is employed to generate the input pulse train. The clock is controlled by a DCM embedded in the Virtex-6 and supply voltages are again controlled and monitored by Texas Instrument's TUSB3210

and Fusion software platform. Image processing is executed both through FPGA and MATLAB. Measurement outputs from the FPGA is probed and captured by Tektronix's TLA7016 Logic Analyzer [95]. Saved output pulse trains are converted to 18-bit output format to process the gamma correction. The two data streams, one passed through MATLAB and the other through FPGA are both ultimately processed in MATLAB to produce the final calculation errors.

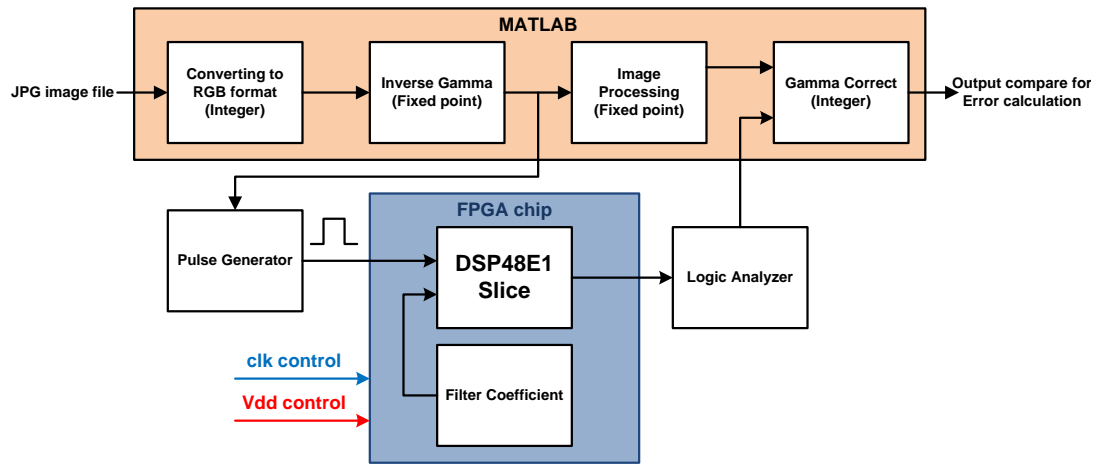


Figure 4.22 Detailed experiment set-up for minimal hardware implementation, which minimize the software dependency of the conventional FPGA tool flows.

Besides the effort to minimize the software dependency, minimized use of soft IP is also pursued by using a custom-coded HDL implementation. In the FPGA chip implementation in Figure 4.22, the only hardware being exploited is the DSP48E1 slice. To get rid of the possibility of unexpected FPGA synthesis completely, the minimal Hard IP blocks and routings of metal wires are checked and confirmed manually. A 2-D 3 tap FIR filter implementation is initiated by hand and converted to the basic building blocks step by step. The 2-D FIR filter is divided into three 1-D 3 tap FIR filters as shown in Figure 4.23.

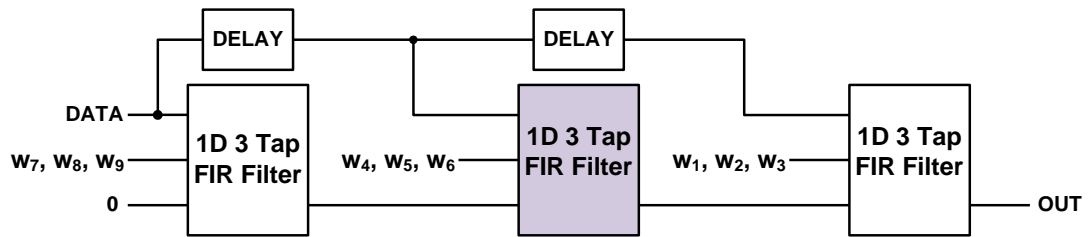


Figure 4.23 Implementation of 2-D 3 tap FIR filter using three 1-D 3 tap FIR filters

Each 1-D 3 tap FIR filter is designed using the conventional signal flow diagram which is widely used for the implementation of filters in DSP applications. The overall implementation of the signal flow diagram is achieved using the two registers and 3 MAC units. This top-down design and checking flow are illustrated in Figure 4.24 and Figure 4.25. Filter coefficients are also provided through external input pins after generating pulse trains using the programmable pattern generator.

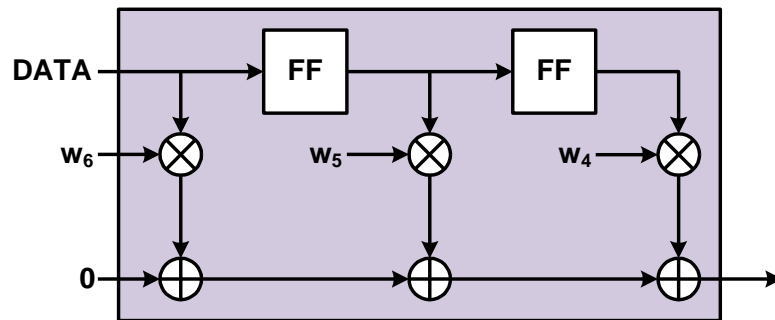


Figure 4.24 Signal flow diagram for 1-D 3 tap FIR filter.

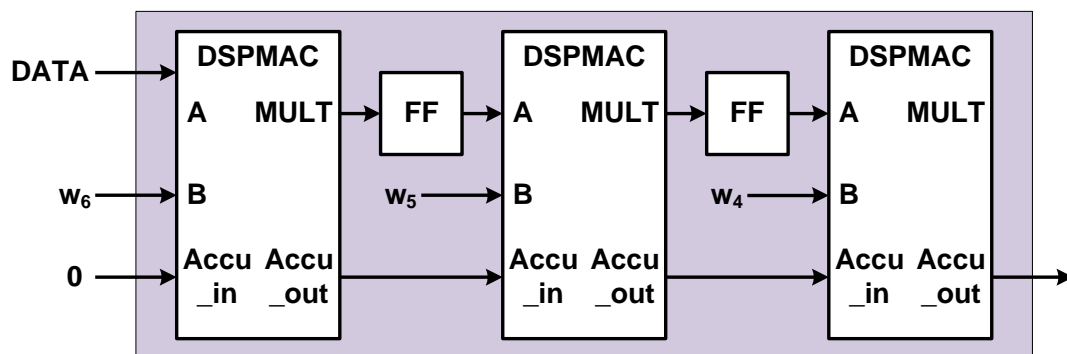


Figure 4.25 Implementation of the above signal flow using the Virtex-6's DSP48E1 macro.

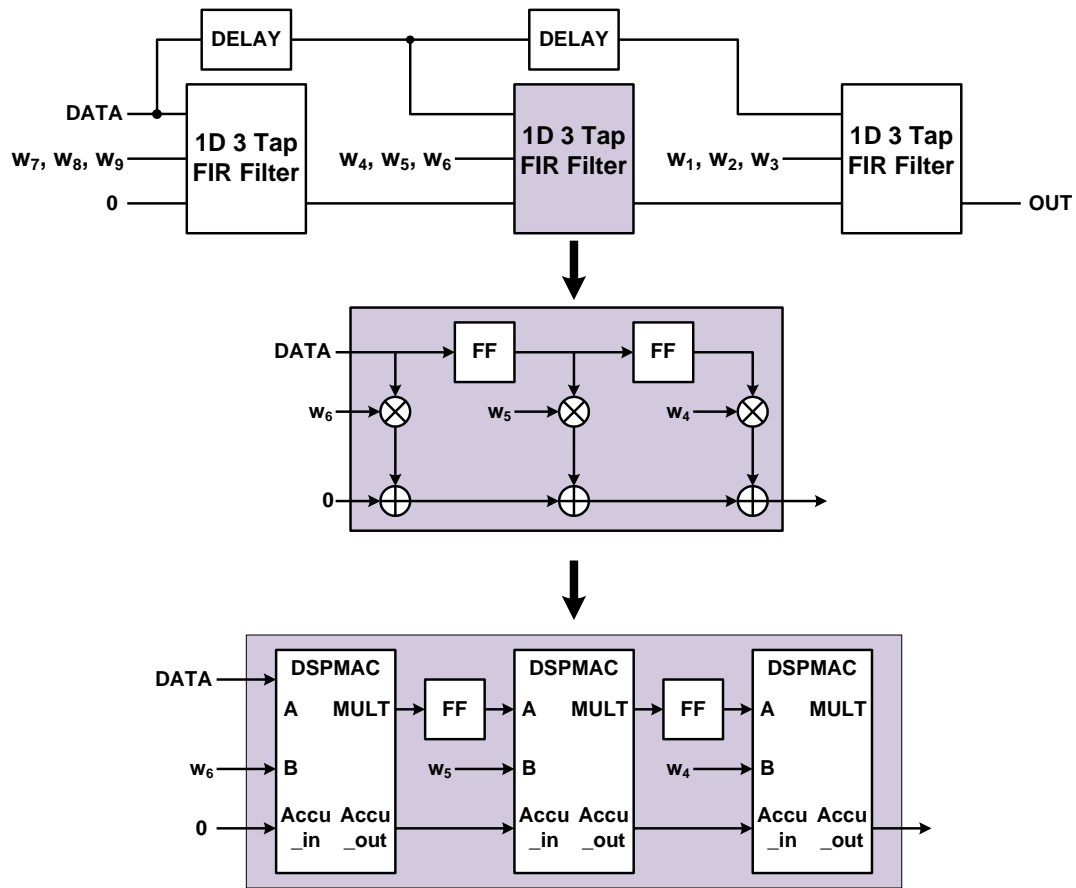


Figure 4.26 Overall implementation flow for 2-D 3 tap image filter for image sharpening.

With the varying clock frequencies suggested by ISE Design Suite's simulation tool, frequencies a bit less than 300 MHz are found to be the maximum operating frequency to process the image with no calculation error. However, depending on the optimization and PVT variations, the maximum frequency fluctuates between 286MHz to 300MHz. Unfortunately, due to the limit of the programmable pulse generator, it was not possible to try the experiments at frequencies above 300MHz. Measurement data is processed for comparison and the final calculation error is plotted in Figure 4.27. Image pixels are scanned from the upper-left corner to lower-right corner, from left to right and up to down. R, G, and B color components are

processed in parallel and collected later to reconstruct the final images. Each experiment is performed 1,000 times using the same original image since a single attempt does not result in enough trials for error detection to be evaluated, especially for the cases with lower error rates. Calculation errors are plotted with respect to normalized power consumption. At each power supply voltage value, average power consumption is monitored and recorded. Figure 4.27 is a scatter plot of the resulting power/calculation error rate data pairs.

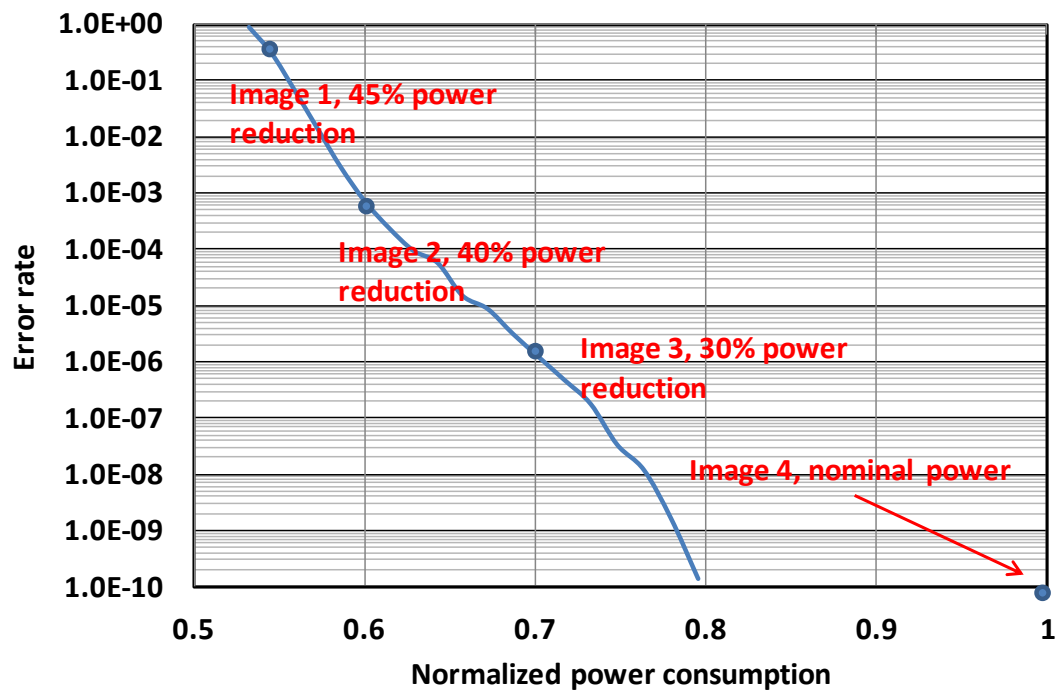


Figure 4.27 Relationship between powers consumption and calculation error. Four reference points are indicated along with the corresponding power numbers. Actual image quality for each of the indicated points is displayed in the next section.

4.4 Measurement Result and Discussion

Reconstructed images are displayed successively in Figure 4.28 to 4.31. Each image shows the degree of quality degradation along with the corresponding possible power reduction. The results reinforce the claim that inexact computing is a promising ultra-low power design methodology. It is indeed possible using this methodology to tune the energy consumption until the quality degradation becomes excessive and intolerable, which in itself will be determined by the specific applications.



Figure 4.28 Image quality is significantly degraded and may not be acceptable. However, note that the reduction in power consumption is 45%.



Figure 4.29 Image with noticeable quality degradation. However, note that power consumption is reduced by 40%, possibly making the scenario acceptable for certain applications.



Figure 4.30 Image with hardly any noticeable quality degradation while reducing power consumption by 30%.

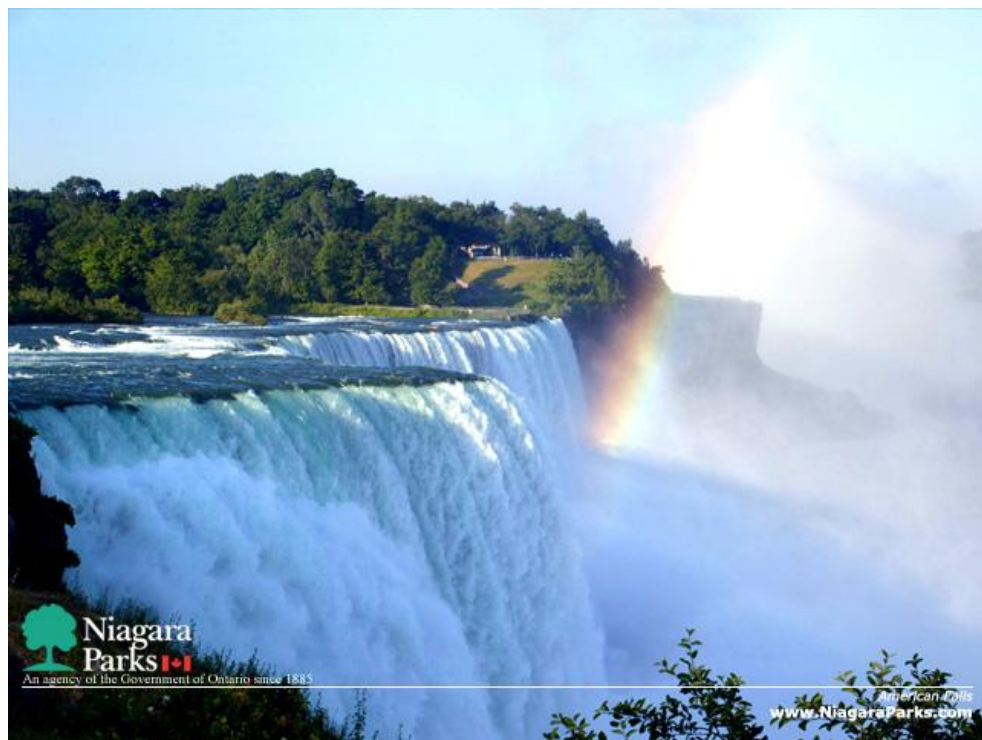
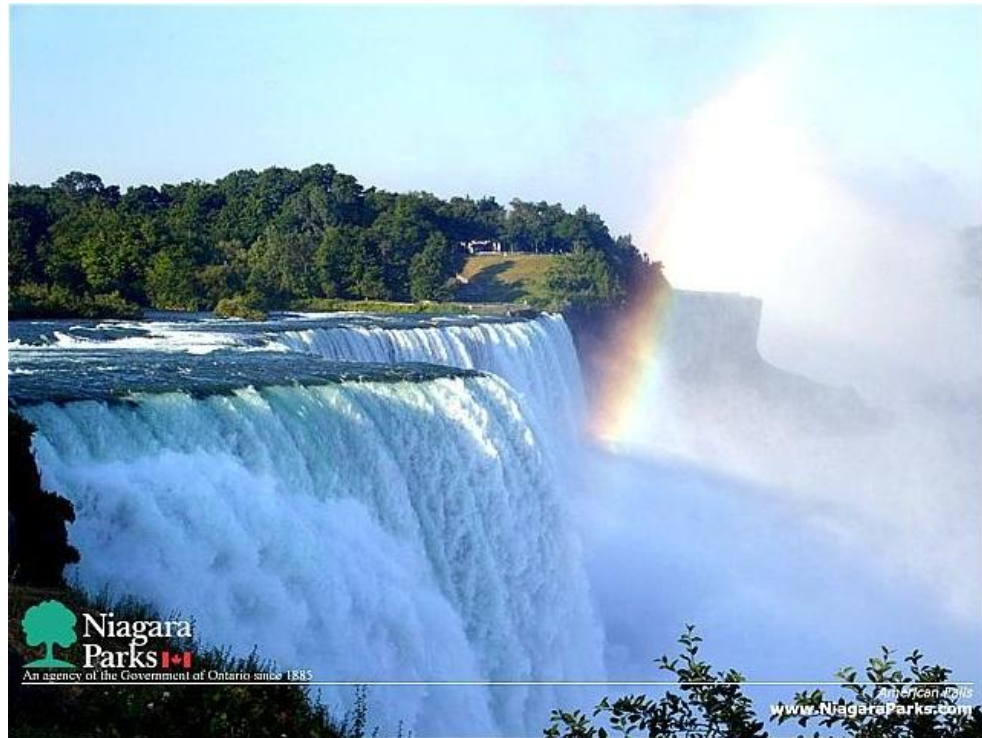


Figure 4.31 Image with no error (lower). Upper image is the same one shown in Figure 4.26 with 30% reduction in power consumption, displayed again for comparison. This is very attractive result for image processors in mobile applications.

4.5 Chapter Summary

In this chapter, the experimental framework to verify our proposed ultra-low power design methodology using ‘Inexact Computing’ on an actual hardware implementation platform was presented. A tradeoff between the quality of contents – in this case a JPEG image – versus power consumption (or extended battery life) is demonstrated. This result indicates a very clear possibility for this inexact computing methodology to be applied to achieve significant reduction in power consumption for applications that can tolerate a specified maximum computational error. By employing the proposed MSB-LSB weighted supply voltage scaling scheme, it may still be possible to continue satisfy the seemingly impossible demands for both reduced power consumption and increased functionality for each generation of electronic devices even in the current era of power-limited scaling. Further applications of this inexact computing and its extension to Many-core architecture [96] and Near-Threshold Computing [97] (NTC) will be presented in the next chapter.

CHAPTER 5

FUTURE RESEARCH DIRECTIONS AND CONCLUSIONS

5.1 Double-Gate MOSFETs and its Adaptive design applications

Although many research groups have shown an implementation of the planar back-gated MOSFET device structure [17, 18] as shown in Figure 5.1, a successful mass-production using this type of device structure has not been reported yet. The recent success in mass-production of Tri-gate devices fabricated using Intel's 22nm technology, however, promises a possible gateway for this independently biased Double-Gated MOSFET structure. As illustrated in Figure 5.2, vertical structure of Tri-gate can be easily transformed to the Fin-Type Double-Gate version by etching the overlapped gate material until the opening of the channel and completely disconnecting the front and back gates.

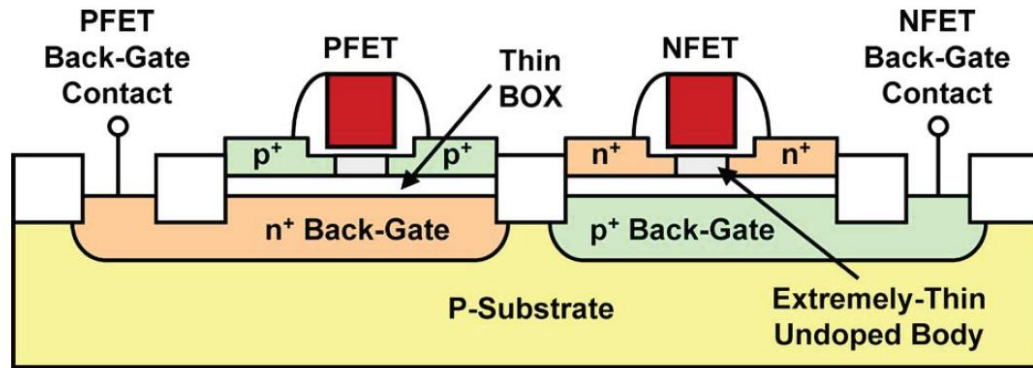


Figure 5.1 An example of fabrication of the planar type back-gated MOSFET structure. NFET and PFET back-gate electrodes are placed beneath of the channels and employed for threshold voltage modulation.

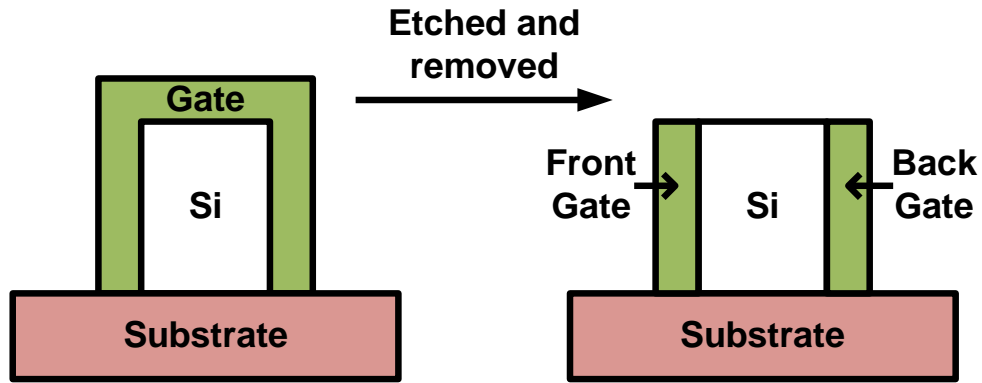


Figure 5.2 Transformation of the device structure from Tri-gate FinFET to Fin-type Double-Gate MOSFET.

Considering one main challenge in Tri-gate fabrication as a control of threshold voltage, the enablement of independently biased back-gated DGMOSFET would be a candidate to address this issue. However, as noted in chapter 2, the iterative algorithm of DGMOSFET is delaying an implementation into a robust compact device model. Therefore, a simpler and efficient analytical approximation for the front and back surface potential would be necessary. In addition, the compact device model has to be strengthened to accommodate the recent progresses in device technology such as strain engineering, high-k metal gate stack, and other enhancements suppressing the pronounced Short-Channel Effects.

Even after the successful development of the robust compact device model, routing a front gate as well as a back gate independently will impose an overhead in terms of layout and integration density. Figure 5.3 shows a possible scenario when doing a layout of Tri-gate FETs. If we want to add a back-gate routing, this will increase the fin pitch inevitably and decrease the layout density, which significantly degrades the merit of high-density circuit blocks such as an embedded SRAM.

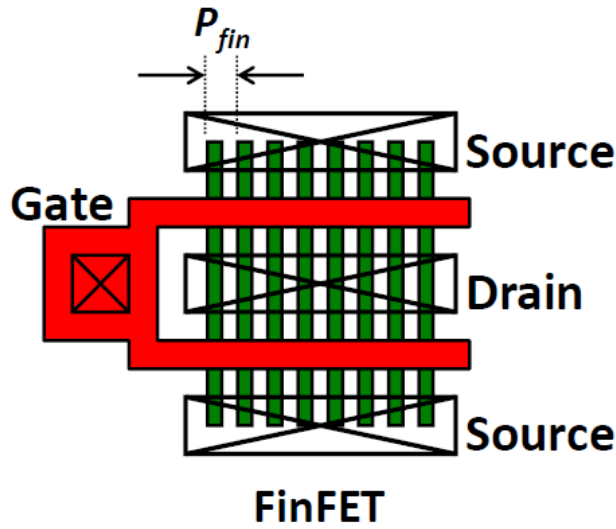


Figure 5.3 Layout of FinFET. Fin pitch should be increased significantly to accommodate the additional routings of metal wires for independently biased-back gates.

However, if the instantiation of the DGMOSFET devices are limited for particular purpose as presented in Figure 5.4, the tradeoff between the layout overhead and performance advantages can be considered again. The two access transistors are independently biased DGMOSFET and the four other transistors in the inverter pair are common-gate DGMOSFET. Figure 5.5 shows the simulated butterfly curves [98] for the 6-T SRAM cell. The significantly improved read margin due to back-gate threshold voltage adjustment is shown.

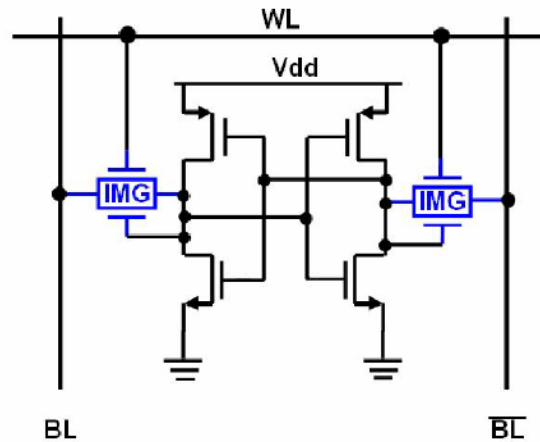


Figure 5.4 Circuit schematic for selective use of DGMOSFET to minimize the layout overhead.

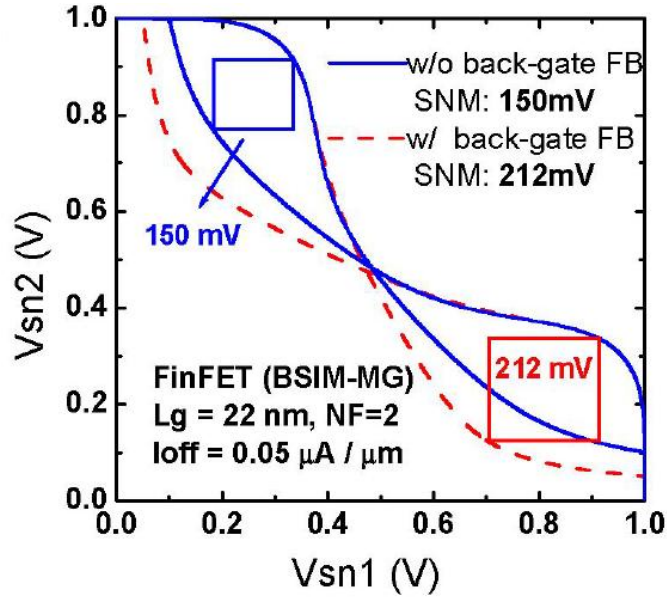


Figure 5.5 Butterfly plots and read margins extraction results.

Similar simpler methodologies can be applied to logic circuit. Our adaptive circuit design using the independently biased back-gated DGMOSFET in chapter 2 assumes the situation that all back-gates are tied together in certain block of interest. This means that our adaptive design technique is viable even to logic circuits while reducing the design overheads. Tradeoff between the performance benefits and the design overhead should be carefully taken into account for this case.

Due to the quantization of widths in FinFET, the optimal sizing of the transistor widths in dense SRAM bitcells is not available any more. Possible configurations with the tradeoffs of robustness vs. density have been proposed such as 1-1-1, 1-2-2, and 1-3-3 configurations. This configuration denotes the number of fins for Pull-UP transistors, for pass-gate transistors, and for Pull-Down transistors, respectively. Careful studies seem to be worthwhile to find out which option is the best between 1-1-1 configuration with R/W assist circuits [99], 1-1-1 configuration with the pass-gate

transistors replaced with independently biased type, and 1-2-2 configuration without additional transistors.

Independently biased back-gated DGMOSFET can be also advantageous when used for analog and RF circuit designs, however, the design of these analog and RF circuits using FinFET are still not verified in terms of productivity and performance merits. Once the problems in this type of circuit design using the FinFET are resolved, the opportunities for adaptation using independently biased back-gated DGMOSFET will be possible again.

5.2 Statistical Simulation Framework using Probabilistic circuits

Due to the increasing variability, the manufactured circuit may not have the performance that designers targeted. Variability-aware design should be taken into place in the early stage of the design flows, in order to ensure functional robustness in the presence of variability. To verify whether the circuit is robust under the influence of variability, a statistical model is provided by foundries as shown in Figure 5.6. This model is more realistic than the conventional corner models and covers the full design space. Foundries typically offer a process with 3-sigma global variation number and the number of local sigma is determined by the designer. For example, to design a 1MB SRAM, 5-sigma is required for the bitcell design as shown in Table 5.1. Then, the target design is verified through Monte Carlo simulation using this sigma numbers. In light of technology scaling, the yield estimation problem is plagued by the inability of Monte Carlo methods in calculating rare event probability since the number of

simulations increases as the integration density increases as in Table 5.1.

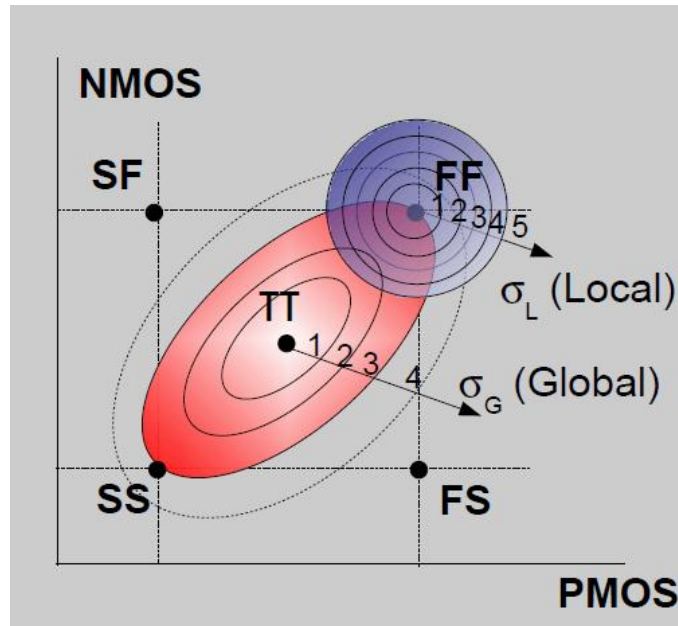


Figure 5.6 Statistical models are provided by foundries for functional robustness under an influence of variability.

Failure criteria:

$$(\mu - N \sigma_G - M \sigma_L) > \text{Safe Margin}$$

where:

- μ is the mean
- N is determined by the foundry and is typically 3.
- M is determined by the number of instances of the circuit being analyzed:

# of instances	M
100	2.33
1,000	3.09
10,000	3.72
100,000	4.26
1,000,000	4.75
10,000,000	5.20

Table 5.1 Failure criteria, which determines how many sigma numbers are required for the target design.

Monte Carlo simulation for rare event probability calculation is consuming more computing power and time, therefore, methods such as importance sampling and convex optimization has been proposed and alleviates the problem, but still requires a significant simulation cost or even becomes worse in case of diverged iterations.

Figure 5.7 shows a comparison between the conventional Monte Carlo simulation with 10k samples of global variations (approximately 4-sigma) at TTG corner and our proposed probabilistic circuit simulation framework. In our framework, five stages of clock buffer which composed of 10 inverters of FO1 are chosen for mimicking timing uncertainty in clock tree. The simulation results are summarized in Table 5.2. For every Monte Carlo run, uncorrelated random seed generates the skewed variations of threshold voltage based on its number of runs (i.e. sigma number). And, these variations in threshold voltages in turn are applied for evaluating the propagation delay. In probabilistic circuit approach, 10k cycles of simulation is performed and each propagation delay which affected by injecting an AWGN noise source at input node is evaluated for normal plot.

	Monte Carlo	Probabilistic	Difference (%)
Mean (ps)	51.0204	51.6085	1.15
+4 sigma value (ps)	71.8184	72.8024	1.37
-4 sigma value (ps)	38.1039	38.483	0.99
Elapsed time (sec)	3014.84	538.72	-82.13
Memory used (MB)	145	142	-2.1

Table 5.2 Comparison between Monte Carlo simulation and Probabilistic circuit approach

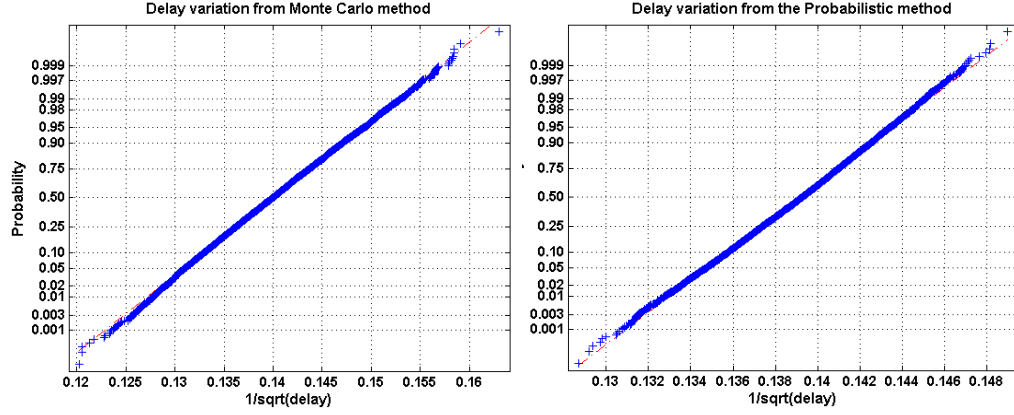


Figure 5.7 Delay variations from the conventional Monte Carlo simulation (Left) and from the probabilistic approach (Right). Both of variations are fitted to normal distribution by taking an inverse of square root of the delay variations. Simulations are based on a commercial 20nm process technology and its statistical models. Note that mean values for both of distributions are almost identical but there is a linear relationship between each standard deviation value. Once a correlation is defined for this linear dependency, the exact same distribution can be acquired from our probabilistic approach.

From the table, we notice that the reduction (5.6X) in simulation time is significant when using our probabilistic approach. Even though the number of samples for this simulation may be relatively small, it suggests a promising approach since the major bottleneck in the statistical approach has been a huge amount of time paid for high-sigma Monte Carlo simulation. Therefore, a further study to elucidate a theoretical background for linking the conventional method with this proposed approach can be worthwhile to pursue. More details on this link are available in Appendix A. Our probabilistic approach for the statistical treatment of random variations would be a better substitute for the time-consuming conventional simulation method, which relied on Monte Carlo in the past. To implement this idea into the EDA tools, software and hardware co-design addressing the proposed probabilistic circuit design methodology will be required.

5.3 Inexact computing: MSB-LSB weighted scaling scheme

In chapter 4, 32-bit MAC was designed using the MSB-LSB weighted scaling scheme, but was not employed for actual image processing hardware. Due to its limited capability of handling fixed-point/floating-point arithmetic operation, the MSB-LSB weighted scheme could not be verified for the enhanced error and energy-efficiency. If we compare the result from Figure 3.18 and 4.27, at the similar calculation error level, MSB-LSB weighted scheme demonstrates the better energy-efficiency than single-supply voltage scaling of the FPGA. The direct comparison may be harder because we have no information of the internal design margin of the FPGA and both of circuits have different topology and architecture. However, assuming that the applied clock and supply voltage were determined based on the 20% of additional design margins, this architectural advantage from the novel supply scaling methodology would extend to the case of general computing examples. As a future research, a modification of the circuit block to support a fixed-point arithmetic operation also would be worthwhile to try.

5.4 Inexact computing: Other application examples

In the previous chapter, we demonstrate the effectiveness of Inexact computing by employing a simple image processing prototype. This prototype filters the original images for edge detection or sharpness enhancement. The processing algorithms for 2D input are not limited on the images, but extended to video by manipulating 2D

image input successively within a specified time duration. For example, H.264 codec is widely used as an international standard for High Definition media format. Detailed explanation of this codec is above the scope of dissertation, but as shown in Figure 5.8, a frame of digital video consists of three rectangular arrays of integer-valued samples and these integer values are manipulated in the same way as done in the 2D image system.

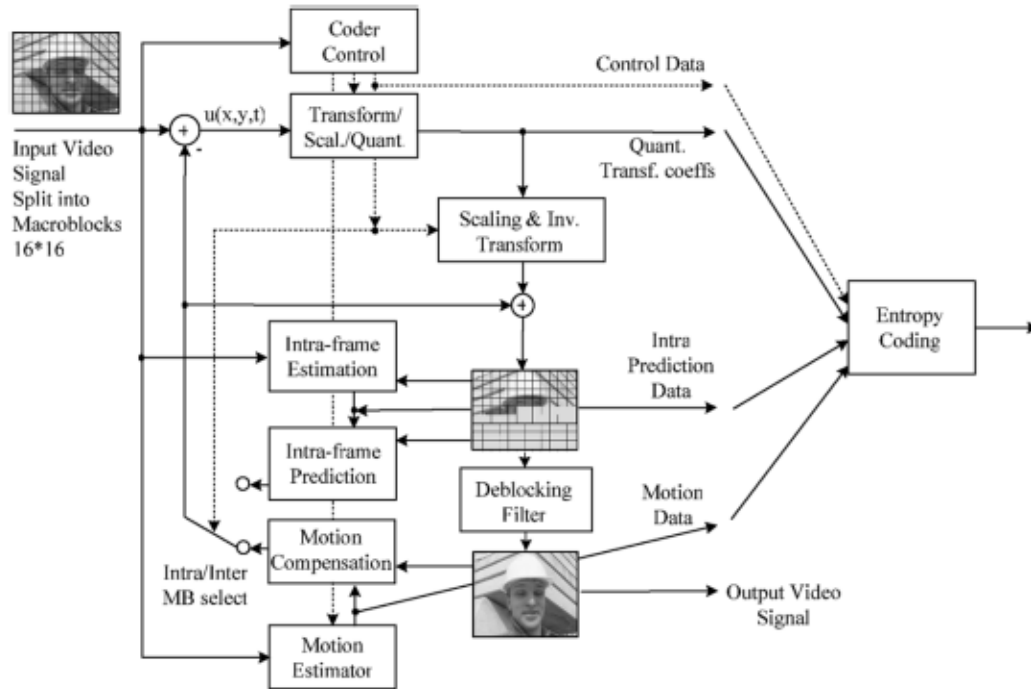


Figure 5.8 Hybrid video encoder (especially for H.264/AVC) [100].

It looks harder to apply the MSB-LSB weighted scaling scheme directly to this complex system, but we can still find many possible opportunities to be more energy-efficient by employing inexact computing. Here one example is introduced.

Video coding often uses a color representation having three components called Y, C_b , and C_r . Y Component is called the luminance component, since it roughly reflects

the luminance l (or *luma*). It is primarily responsible for the perception of the brightness of a color image, and can be used as a black-and-white image [101]. The C_b and C_r components are called chrominance (*chroma*) components, and they are primarily responsible for the perception of the hue and saturation of a color image. Because the human visual system is more sensitive luma than chroma, often a sampling structure is used in which the chroma component arrays each have only one-fourth as many samples as the corresponding luma component array (half the number of samples in both the horizontal and vertical dimensions). This is called 4:2:0 sampling. As one example of Inexact computing, we can provide a higher supply voltage to the luma component processing circuit and lower voltages to the chroma component processing block. In addition, since luminance Y can be represented as the following,

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

different voltage values for R , G , and B component processing can be applied for the implementation of Inexact computing. This example also suggests one possible scenario to exploit the advantage of Inexact computing on real-world computing. Another example of Inexact computing in video signal processing is transmitting its over wireless networks. Since there are less important sections of the video packet, importance-oriented selective treatments of data or a reduction of the number of transmitted bits could be used without affecting the image quality significantly [102].

Elevating our hierarchy to the system-level, non-deterministic or probabilistic decisions on the system can be one possible example. Recently, the concept of Many-core has been introduced by computer architecture communities as one solution to

obtain sustainable performance improvement by employing a parallelism. On a many-core processor [103], surplus transistors are used to replicate simple execution cores across the chip, thereby providing large number of hardware contexts that can execute in parallel; using simple cores can result in better energy efficiency and lower verification cost as well. As a result, the many-core approach has been adopted as the de facto standard: 64-core processors are available in the market [104], and major vendors are following the trend [105]. Both industry experts [95] and academia [106, 107] agree that scaling the number of cores to hundreds or thousands is the only way to scale performance.

Along with the emergence of many-core architecture, the Near-Threshold Computing (NTC) has been also highlighted as a pair to address the power consumption issue. If the number of cores increases to hundreds or thousands, extremely scaled low voltage operation becomes inevitable. However, three key challenges comes with NTC: 1) 10X or greater loss in performance, 2) 5X increase in performance variation, and 3) 5 orders of magnitude increase in functional failure rate of memory as well as increased logic failures [108].

At this point, our inexact computing methodology based on the probabilistic representation of circuit functionality gains a significant importance as a solution to the issues. Due to the 3 key challenges mentioned above, all circuit elements on the chip cannot operate correctly. We should determine a correct functionality from the overall behavior of the chip. Our work in the dissertation may be the first step on its forward way. Systematic methodologies should be created to determine the chip functionality based on the probability and Operating system/software also should

support this probability based approach. Co-design of hardware and software are becoming more critical in achieving this design goal.

As one possible implementation example of this probabilistic approach, the circuit in Figure 5.9 is proposed. Instead of placing ring-oscillators at several points of interest in the chip to monitor the local performance and variation impact, the sensors for detecting the probability of errors can be placed locally. Random noise is generated by Fibonacci ring oscillator with restarting control [109] and injected to one of the inverter inputs. Phase detector outputs a difference between the inverter outputs with and without input noise signals, respectively. Low pass filter's output level is determined by how different between the two of inverter outputs. These sensors give their outputs, which corresponds to the probability of failure, to the central controller to adjust the system parameters such as supply voltages and frequencies.

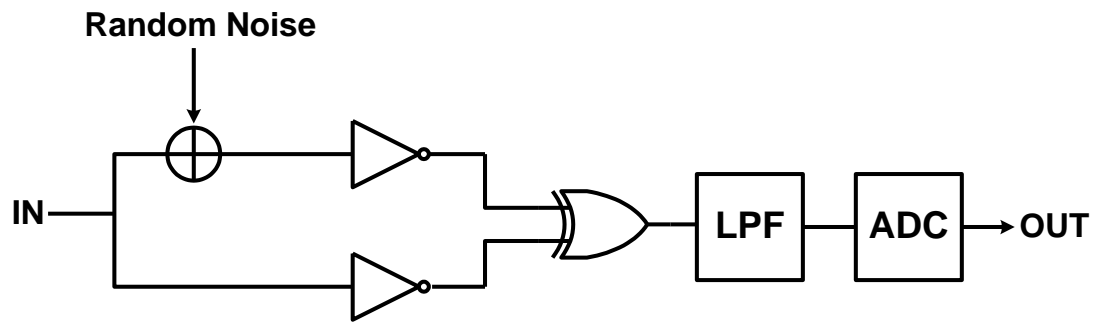


Figure 5.9 Sensor implementation for detecting error probability. Test input propagates two inverters, one without noise and the other with noise. Based on the relationship found in the previous chapters, the output voltage level can be adjusted for determining the tolerable level of error probability.

5.5 Conclusions

Silicon-based CMOS technology still has a potential for continued dimensional shrinkage into the sub-10nm regime, but design methodologies for power- and variability-limited device miniaturization are needed to exploit the benefits of shrinking transistor dimension. A new adaptive circuit design approach using DGMOSFET devices is proposed as a solution to address the above problems and this proposed adaptation strategies would allow circuit designers to produce variation-free, high noise-margin circuits with workload awareness that allows for flexible tradeoff between power consumption and performance at sub-50 nm technology nodes. Another approach to low-energy computation using statistical performance metrics that incorporate error margins as a constraining requirement is elaborated. Utilizing the probability of error as an allowable design tolerance, we show that a simultaneous optimization of both energy-efficiency and computing error is possible. This provides the circuit designer with greatly increased flexibility to trade off between energy and calculation accuracy. As an application of this concept, a simple image processing prototype implemented on a FPGA chip shows the potential benefits of inexact computing. A tradeoff between the quality of contents versus power consumption (or extended battery life) is also demonstrated using this prototype. This promises a clear possibility for this Inexact Computing methodology to be applied to general computing domain.

APPENDIX

A. Probabilistic approach for Statistical representation of delay variations

In Chapter 3, the effects of random noise injected into circuit nodes and the dominance of input-coupled noise were discussed. There, it was found that random noise modeling uncertainty in nanometer VLSI circuits results in timing variations. Especially in synchronous digital design, since this timing variation impacts its critical path delay and degrades overall circuit performance, a timing guardband generated from statistical methodology has to be employed in early stage of design. Additional representations of treating timing variation as an error instead of voltage difference will be introduced in this section. Also, in chapter 5, two delay distributions from the clock tree were shown for a comparison of conventional Monte Carlo vs. probabilistic approach. A quantitative analysis behind this comparison will be elaborated here to suggest a new approach which may replace time-consuming conventional Monte Carlo methodology.

A clock network-only analysis can simplify our approach to show equivalence between conventional Monte Carlo and the probabilistic method. By limiting the analysis to clocks, theoretical modeling is only required for the clock cells, i.e. inverters, which is typically the simplest form of the digital circuit. Since the clock tree is one of the most variation-sensitive parts of the design [110], our approach only considering clock tree would capture the timing variation in full-chip library. As shown in Figure A-1, five stages of clock buffers are employed to verify our approach.

Each clock buffer is composed of a pair of inverters of FO1 and cascaded to form a path of clock tree. We used a commercial 20nm bulk process technology and its HSPICE statistical model provided by foundry. Then, an input signal is applied to the clock tree and its propagation delay is evaluated by capturing the signals at output node. As we did in chapter 3, the band-limited noise sampling is used again to guarantee that most of input signals to be propagated without filtering which depends on the frequency response of the devices.

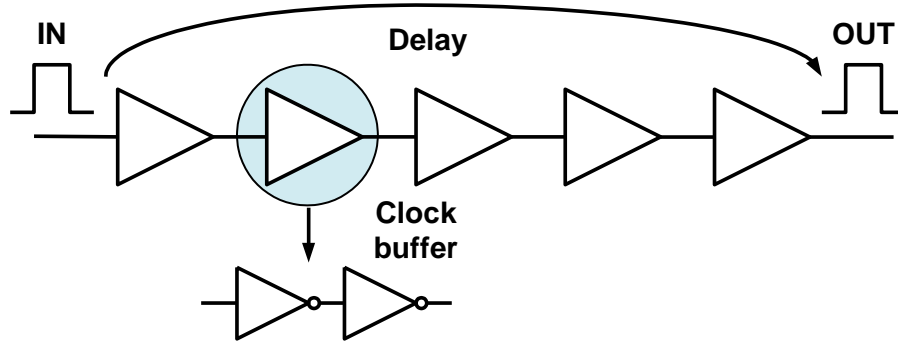


Figure A.1 A clock tree of five stages clock buffers. A clock buffer is composed of a pair of inverters of FO1. Propagation delay is evaluated for Monte Carlo simulation and probability method.

The propagation delay of the inverter can be represented as an integral form of the capacitor (dis)charge current as the following.

$$t_p = \int_{v_1}^{v_2} \frac{C_L(v)}{i(v)} dv \quad (A.1)$$

i is the (dis)charging current, v the voltage over the capacitor, and v_1 and v_2 the initial and final voltage, respectively. A direct calculation of above equation is complicated, since both $C_L(v)$ and $i(V)$ are nonlinear functions of v . Instead, we use the simplified switch model of the inverter introduced in Figure A.2 to derive a reasonable approximation of the propagation delay. The voltage dependencies of the “on”

resistance and the load capacitor are treated as a constant linear element with a value averaged over the interval of interest. An expression for the average “on” resistance of the transistor was already derived in [38] as follows.

$$R_{eq} = \frac{1}{V_{DS}/2} \int_{V_{DS}/2}^{V_{DS}} \frac{V}{I_{DSAT}(1 + \lambda V)} dV \approx \frac{3}{4} \frac{V_{DS}}{I_{DSAT}} \left(1 - \frac{7}{9} \lambda V_{DS}\right) \quad (A.2)$$

$$I_{DSAT} = \mu C_{ox} \frac{W}{L} \left((V_{GS} - V_{th})V_{DS} - \frac{V_{DS}^2}{2} \right)$$

From a first-order linear RC analysis, the propagation delay is

$$t_{pHL} = \ln(2)R_{eqn}C_L = 0.69R_{eqn}C_L, \text{ similarly, } t_{pLH} = 0.69R_{eqp}C_L \quad (A.3)$$

If we size the P/N ratio to have the same transitioning time, then high-to-low and low-to-high transitions are identical. The overall propagation delay of the clock tree is defined as propagation delay of one inverter stage multiplied by the number of inverter stages, n :

$$t_{delay} = nt_p = 0.69nC_LR_{eq} \quad (A.4)$$

Now we have an equation for the delay, it's time to look at the two different distributions from the conventional Monte Carlo simulation as well as from our proposed probabilistic method as shown in Figure A.3. Upper figure is a normal plot from the foundry's global variation HSPICE model. 10k of Monte Carlo simulations were executed by applying V_{DD} of 0.9V at TTG corner. 10k of samples is approximately equivalent to four sigma number. The lower one is a normal plot from our proposed probabilistic simulation framework. 10k of input streams are pipelined to the clock tree and the propagated delays are evaluated at every cycle. Due to the differences of random overridden voltages on input at every cycle, the resulted

propagation delay would be different for every cycle and have a certain distribution function. Simulation is also performed with V_{DD} of 0.9V at TT corner.

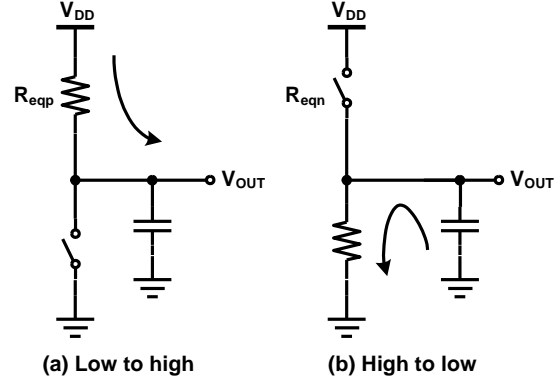


Figure A.2 Simplified switch model of dynamic behavior of static CMOS inverter.

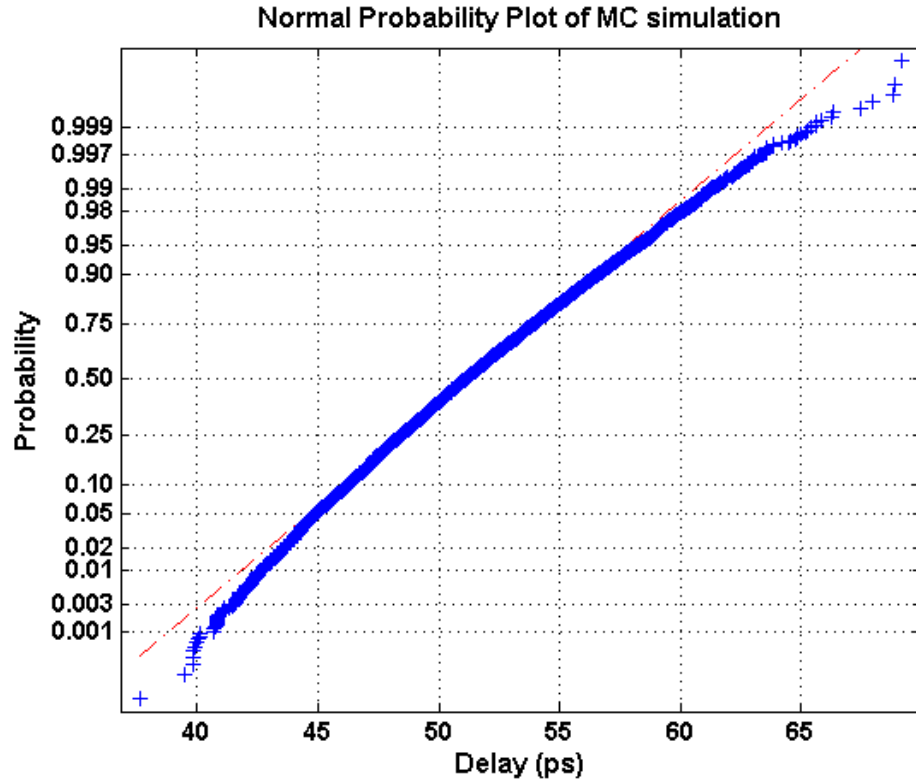


Figure A.3 Delay distribution function from the 10k of Monte Carlo runs of the clock tree. Mean is 51.3 ps and starts deviating from the normal distribution around 2 sigma.

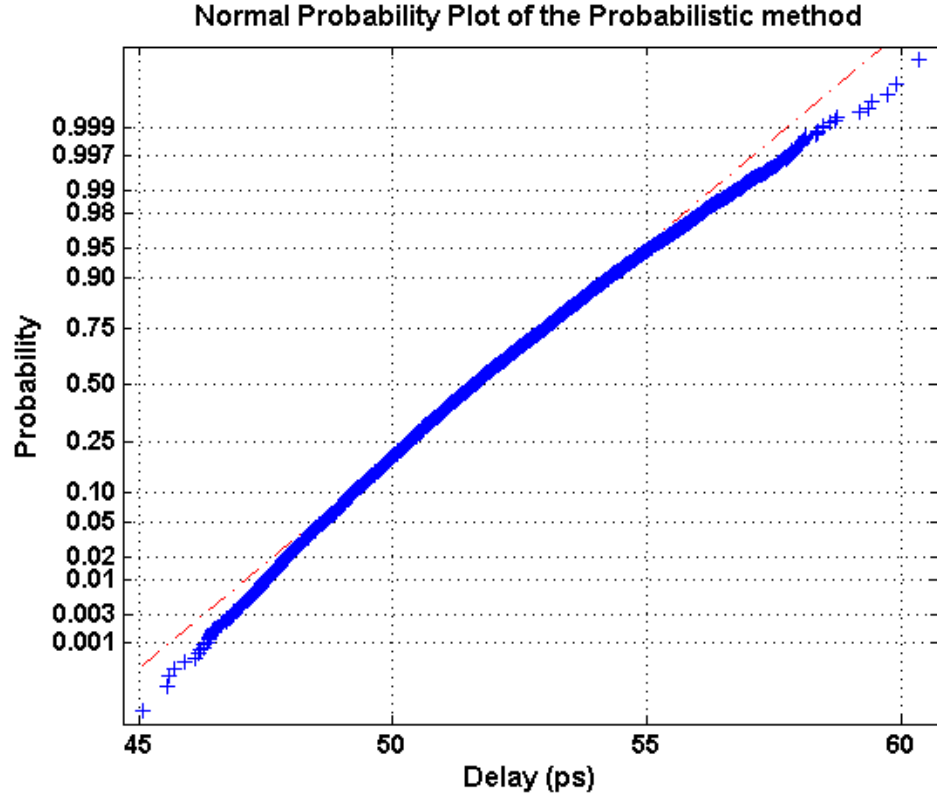


Figure A.4 Delay distribution function from the proposed probabilistic approach. Mean is 51.6 ps and starts deviating from the normal distribution around 2 sigma.

Considering the fact that in the foundry's statistical model, the random Gaussian functions are applied to the threshold voltage and this is the most significant sources of variations in modern nanometer process technologies, the equations of A.2 and A.4 can be re-written for the following relation between the threshold voltage and the propagation delay

$$t_{delay} \approx 0.26n \frac{V_{DD} \left(1 - \frac{7}{9} \lambda V_{DD}\right)}{\mu C_{ox} \frac{W}{L} (V_{GS} - V_{th})^2} \quad (A.5)$$

where transistor is assumed to be mostly in saturation region due to V_{DD} of 0.9V, and V_{DS} is replaced with V_{DD} .

This in turn means

$$(V_{GS} - V_{th}) \propto \frac{1}{\sqrt{t_{delay}}} . \quad (A.6)$$

Since V_{GS} is a constant in the Monte Carlo simulation and V_{th} is a Gaussian distribution, $V_{GS} - V_{th}$ is again a Gaussian distribution. This relation is verified by taking inverse of square root of the delay variations as shown in Figure A.5.

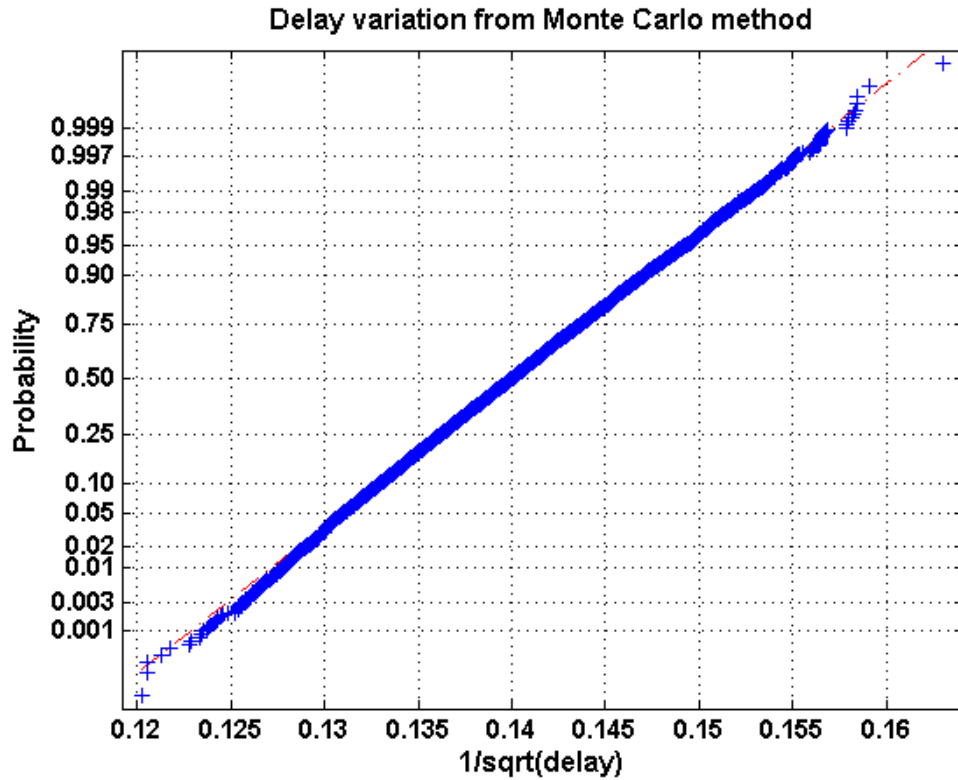


Figure A.5 $1/\sqrt{\text{delay}}$ is observed to be a perfect Gaussian distribution and this verifies our simplified delay modeling in this section. Since the distribution is a perfect Gaussian, we can get more accurate sigma variation values by linear extrapolation.

Again, the same fitting is applied to the delay variations from the probabilistic approach as shown in Figure A.6.

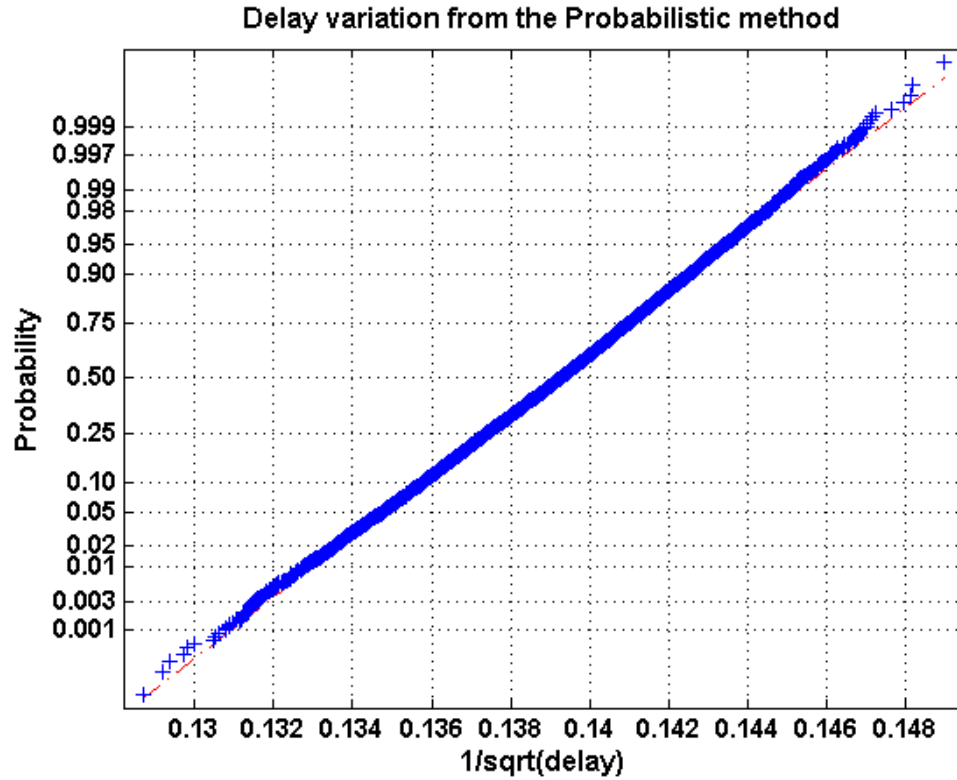


Figure A.6 $1/\sqrt{\text{delay}}$ fitting is also applied to the delay variations from the probabilistic method. Note that the same perfect Gaussian distribution is acquired through this fitting and this proves that our suggested probabilistic method is identical to the result from the Monte Carlo simulation.

Since the two delay variations from the different approaches have the same tendency, certain correlation coefficient between the two variations can be found by this linearity. For example, in the Figure A.5 and A.6, only difference is a steepness of each normal distribution, which is the difference in standard deviation values, σ . By applying the ratio (2.082 in this case) of each sigma - this ratio depends on the characteristics of the process technologies - to the probabilistic method, the results shown in the Table 5.2 is found.

B. Probabilistic methodology for Statistical variations: Simulation details and how to

In previous section, the theoretical equivalence of the traditional statistical to our proposed probabilistic methodology was discussed. More details in the probabilistic simulation methodology will be provided here to help a better understanding of this approach.

The most important thing in the probabilistic simulation framework is how to define the noise sources on input node and how to override this noise on to the input signal streams. As stated in the previous section, the most dominant variation in the process technologies is the threshold voltage variation. To have the same variation effects on the threshold voltages, a random Gaussian noise with zero mean and standard deviation of 50mV is overridden on the input signals and the actual threshold voltage is fixed during this simulation. 50 mV is chosen for the initial standard deviation value for this experiment since the standard deviation of the threshold voltage variation from the Si data is around 50 mV [111]. Assuming ΔV as a random variable of a Gaussian distribution in threshold voltage, $(V_{GS} - V_{th})$ in the equation A.5 is equivalent to both cases as

$$(V_{GS} - V_{th}) = V_{GS} - (V_{th0} - \Delta V_1) = (V_{GS} - \Delta V_2) - V_{th0} \quad (B.1)$$

where ΔV_1 is the randomness provided by the foundry's statistical model, ΔV_2 the randomness from the noise source in our probabilistic platform, and V_{th0} is a constant threshold voltage in case of no process variation effect. In addition, if we make the random noise voltages to be kept during each cycle as shown in Figure B.1, we can

perfectly mimic the situation where the traditional Monte Carlo method was executed (the randomly selected threshold voltage value is not changed during each Monte Carlo run) and reduce the effect of transient circuit dynamics. Also, as become clear in chapter 3, the appropriately chosen band-limited noise source can minimize the effect of dynamic characteristics of the circuits such as filtering and amplifying effects. In this simulation, 2.5 GHz of random Gaussian noise sources are employed along with the minimal transitioning time between the different noise values.

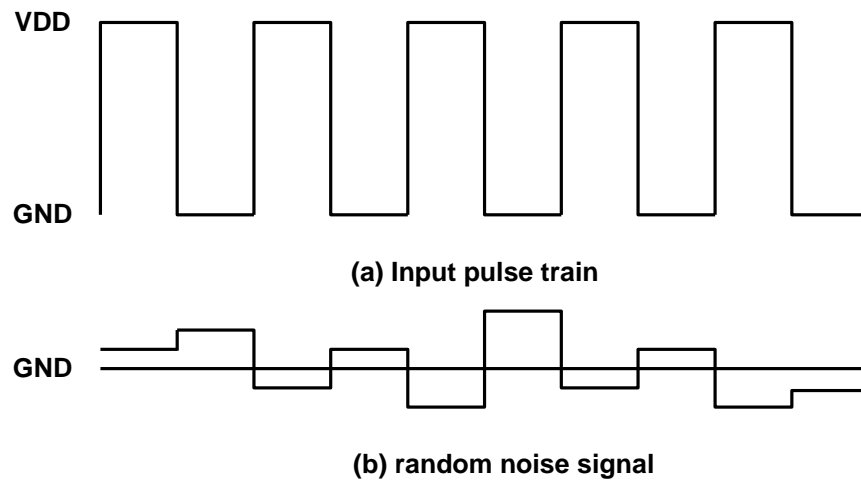


Figure B.1 Random noise signal to mimic the situation where the traditional Monte Carlo method was employed. To minimize the dynamic characteristics of the circuits, transitioning time between the noise voltages are reduced as least as possible. In this simulation, slew is 50 ps.

BIBLIOGRAPHY

- [1] T. Ghani, "Challenges and Innovations in Nano-CMOS Transistor Scaling," Intel, Oct. 2009.
- [2] Y. Nuevo, "Cellular phones as embedded systems," Digest of Technical papers ISSCC, pp. 32-37, Feb. 2004.
- [3] S. Wu et al., "A Highly Manufacturable 28nm CMOS Low Power Platform Technology with Fully Functional 64Mb SRAM Using Dual/Triple Gate Oxide Process," Digest of Technical papers Symp. VLSI Tech., pp. 210-211. June 2009.
- [4] T. B. Hook et al., "Lateral ion implant straggle and mask proximity effect," IEEE Trans. on Elec. Dev., Sep. 2003, pp. 1946-1951.
- [5] T. Ghani et al., "A 90nm Volume Manufacturing Logic Technology Featuring Novel 45nm Gate Length Strained Silicon CMOS Technology," Technical Digest, IEDM 2003, pp. 11.6.1-11.6.3, Dec. 2003.
- [6] International Technology Roadmap for Semiconductors, 2011 Edition.
- [7] M. Qazi et al, "Challenges and Directions for Low-Voltage SRAM," IEEE Design & Test of Computers, Vol. 28, pp. 32-43, Jan.-Feb. 2011
- [8] E. J. Nowak, "Maintaining the benefits of CMOS scaling when scaling bogs down," IBM J. Res. Dev., vol. 46, pp. 169–180, Mar./May 2002.
- [9] D. J. Frank, R. H. Dennard, E. Nowak, P. M. Solomon, Y. Taur, and H.-S. P. Wong, "Device scaling limits of Si MOSFETs and their application dependencies," Proc. IEEE, vol. 89, pp. 259–288, Mar. 2001.
- [10] U. Avci and S. Tiwari, "Back-Gated MOSFETs with Controlled Silicon Thickness MOSFETs for Adaptive Threshold Voltage Control," Electronics Letters, vol. 40, No. 1, pp. 74-75, 2004.

- [11] K. Nose, and T. Sakurai, "Optimization of VDD and VTH for Low-Power and High-Speed Applications," in Proc. Of ASPDAC, pp. 469-474, Jan. 2000.
- [12] G. Gammie et al., "SmartReflex Power and Performance Management Technologies," Proc. of the IEEE, vol. 98, no. 2, Feb. 2010.
- [13] J. Tschanz et al., "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," ISSCC Dig. Tech. papers, Feb. 2007, pp. 292-293.
- [14] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De, "Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs," Proc. LPED'01, pp. 207-212, Aug. 2001.
- [15] D. Frank, S. Laux, and M. Fischetti, "Monte Carlo simulation of a 30nm dual-gate MOSFET: How far can Si go?" in *1992 IEEE Int. Electron Devices Meeting Tech. Dig.*, San Francisco, CA, p. 553.
- [16] I.Y. Yang, C. Vieri, A. Chandrakasan, and D.A. Antoniadis, Back-gated CMOS on SOIAS for dynamic threshold voltage control, *IEEE Trans. Electron. Dev.* 44 (5) (1997) 822-831.
- [17] H. Lin, H. Liu, A. Kumar, U. Avci, J. S. Van Delden and S. Tiwari, "Strained Si Channel Super-Self-Aligned Back-Gate/Double-Gate Planar Transistors," *IEEE Electron Device Letters*, vol. 28, p. 506, 2007.
- [18] K. W. Guarini et al., "Triple-self-aligned, planar double-gate MOSFETs: devices and circuits," *Electron Devices Meeting, IEDM Technical Digest.* P19.2.1 – 19.2.4, 2001.
- [19] P. M. Solomon, "Physically Based, Mixed Mode Compact Model for the Double Gated FET," unpublished 2000.
- [20] M. Jeong, H. S. P. Wong, Y. Taur, P. Oldiges and D.J. Frank, "DC and AC performance analysis of 25 nm symmetric/asymmetric double gate, back gate and bulk CMOS," 2000 International Conference on Simulation Semiconductor Processes and Devices, pp.147-150.

- [21] C. Sampedro et al, "Multi-Subband Monte Carlo simulation of bulk MOSFETs for the 32nm-node and beyond," IEEE ESSDERC Conf. 2010, pp238-241, Sep. 2010
- [22] M. Poljak et al, "Modeling study on carrier mobility in ultra-thin body FinFets with circuit-level implications," IEEE ESSDERC Conf. pp242-245, Sep. 2010
- [23] H. S.P. Wong, D.J. Frank, and P.M Solomon, "Device design considerations for double gate, ground plane, and single gated ultra-thin SOI MOSFET's at the 25 nm channel length generation," IEDM Technical Digest, pp.1407-410, 1996.
- [24] Atlas User Manual, Sep, 2010, Silvaco.
- [25] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Numerical Recipes, the Art of Scientific Computing," London, Cambridge University Press, 1986.
- [26] W. Liu, "MOSFET Models for SPICE Simulation including BSIM3v3 and BSIM4," John Wiley & Sons, 2001.
- [27] See <http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html> for more information and latest development.
- [28] R. Aisola et al, "Verilog-A Language Reference Manual Version 1.0," Open Verilog International, Aug. 1996.
- [29] A. Hokazono, K. Ishimaru, C. Hu, and T -K. Liu, "Forward Body Biasing as a Bulk-Si CMOS Technology Scaling Strategy," IEEE Trans. Electron Devices, Vol. 55, No. 10, Oct. 2008.
- [30] A. Drake, R. Senger, H. Deogun, G., Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala, "A Distributed Critical-Path Monitor for a 65 nm High-Performance Microprocessor," ISSCC, 11-15 Feb 2007, pp. 398-399.
- [31] M. Elgebaly and M. Sachdev, "Variation-Aware Adaptive Voltage Scaling System," IEEE Trans. On VLSI Systems, vol. 15, no. 5, May 2007, pp. 560-571.

- [32] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda, T. Sakurai, "Variable Supply-Voltage Scheme for Low-Power High-Speed CMOS Digital Design," *JSSC*, vol. 33, no. 3, March 1998, pp. 454-462.
- [33] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-voltage CMOS," *IEEE JSSC*, Vol. 30, No. 8, Aug. 1995.
- [34] K. Nose, M. Hirabayashi, H. Kawaguchi, S. Lee, and T. Sakurai, " V_{TH} -hopping scheme to reduce subthreshold leakage for low power processors," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 413-419, Mar. 2002.
- [35] C. H. Kim and K. Roy, "Dynamic V_{TH} scaling scheme for active leakage power reduction," in *Proc. Design, Automation and Test in Europe Conf. and Exhibition*, 2002, pp. 163-167.
- [36] J. Tschanz, J. T. Kao, S. G. Narendra, R. Nair, D. A. Antoniadis, A. P. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, Nov. 2002.
- [37] T. Kobayashi and T. Sakurai, "Self-adjusting threshold-voltage scheme (SATS) for low-voltage high-speed operation," *Proc. CICC'94*, pp. 271-274, May 1994.
- [38] J. Rabaey, A. Chandrakasan, and B. Nikolic, "Digital Integrated Circuits - A Design Perspective," 2nd Ed. Prentice Hall, 2003.
- [39] J. Lohstroh, "Worst-case static noise margin criteria for logic circuits and their mathematical equivalence," *IEEE Journal of Solid-State Circuits*, vol. SC-18, no. 6, pp. 803-806, 1983.
- [40] M. Liu, M. Cai, and Y. Taur, "Scaling Limit of CMOS Supply Voltage from Noise Margin Considerations," *Intl. Conf. on Simulation of Semiconductor Processes and Devices*, pp. 287-289, 2006.
- [41] J. Kim, P. Solomon, and S. Tiwari, "Adaptive Circuit Design Using Independently Biased Back-Gated Double-Gate MOSFETS," *IEEE Transactions on Circuit and Systems I*, vol. 59, no. 3, March 2012

- [42] S. Das, et al., "Razor II: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE J. Solid-State Circuits*, pp. 32--48, Jan. 2009.
- [43] K. Bowman, J. Tschanz, N. Kim, et al., "Energy-Efficient & Metastability-Immune Timing-Error Detection and Instruction Replay-Based Recovery Circuits for Dynamic Variation Tolerance," *IEEE Journal of Solid State Circuits (JSSC)*, Jan 2009.
- [44] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *Proc. IEEE VLSI Test Symp.*, Apr. 1999pp. 86–94.
- [45] S. Das et al., "A self-tuning DVS processor using delay-error detection and correction," *IEEE J. Solid-State Circuits*, pp. 792–804, Apr. 2006.
- [46] M. Deering, "The Limits of Human Vision," 2nd International Immersive Projection Technology Workshop, 1998.
- [47] S. Narayanan, et al., "Computation as Estimation: A General Framework for Robustness and Energy Efficiency in SoCs," *IEEE Trans. Signal Processing*, vol. 58, no. 8, pp. 4416–4421, Aug. 2010.
- [48] J. Kim and S. Tiwari, "Inexact Computing for Ultra Low-power Nanometer Digital Circuit Design," in *Proc. IEEE/ACM Int. Symp. on Nanoscale Architectures*, pp. 24-31, June 2011.
- [49] H. Poor, "An introduction to signal detection and estimation," New York, NY, Springer-Verlag, 1994.
- [50] K. V. Palem: *Proc. Int. Symp. Verification (Theory and Practice)*, 2003, p. 524.
- [51] J. A. Goguen, "The Logic of Inexact Concepts," *Synthese* 19, Dordrecht, Holland, D. Reidel Publishing, pp. 325-373, 1969.
- [52] K.-U. Stein, "Noise-induced error rate as a limiting factor for energy per operation in digital ICs," *IEEE J. Solid-State Circuits*, vol. 12, no. 5, pp. 527–530, Oct. 1977.

- [53] P. Korkmaz, B. E. S. Akgul, and K. V. Palem, "Ultra-low energy computing with noise: Energy-performance-probability tradeoffs," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, Mar. 2006, pp. 349–354.
- [54] T. Hiramoto et al., "Statistical advantages of intrinsic channel fully depleted SOI MOSFETs over bulk MOSFETs," in Proc. CICC, pp. 19-21. Sep. 2011.
- [55] S. Lee et al., "Record RF performance of 45-nm SOI CMOS technology," in International Electron Devices Meeting Tech. Dig., 2007, pp. 255-258.
- [56] U. Ko et al., "A Self-timed Method to Minimize Spurious Transitions in Low Power CMOS Circuits," in Proc. IEEE Symp. on Low Power Electronics, 1994, pp. 62-63.
- [57] C. Mead and L. Conway, Introduction to VLSI systems, Addison-Wesley, 1980.
- [58] J. Kim et al., "Scale changes in electronics: Implications for nanostructure devices for logic and memory and beyond," J. Solid-State Electronics, vol. 84, pp. 2-12, June 2013
- [59] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," in Proc. IEEE Symp. Low Power Electronics and Design, pp. 3-8. 1995.
- [60] M. Igarashi et al., "A low-power design method using multiple supply voltages," in Proc. IEEE Symp. Low Power Electronics and Design, pp. 36-41. Aug. 1997.
- [61] S. K. Mathew et al., "High-performance energy-efficient dual-supply ALU design," High-performance Energy-efficient Microprocessor Design, pp. 171-187, Springer, 2006
- [62] Jan Rabaey, "Low power design essentials," pp. 271-279, Springer, 2009
- [63] D. Ernst et al., "RAZOR: Circuit-level correction of timing errors for low-power operation," IEEE Micro, pp. 10-19, Nov., 2004

- [64] D. Bull et al., "A power-efficient 32 bit ARM processor using timing-error detection and correction for transient-error tolerance and adaptation to PVT variation," IEEE J. Solid-State Circuits, vol. 46, no. 1, pp.18-31, Jan. 2011
- [65] A. Booth, "A signed binary multiplication technique," Quarterly J. Mechanics and Applied Mathematics, vol. IV, pt. 2, Jun. 1951, pp. 236-240.
- [66] O. MacSorley, "High-Speed arithmetic in binary computers," Proc. IRE, vol. 49, pt. 1, Jan. 1961, pp. 67-91.
- [67] A. Burks, H. Goldstine, and J. von Neuman, "Preliminary discussion of the logical design of an electronic computing instrument, part 1, vol. 1, Inst. Advanced Study, Princeton, NJ, 1946
- [68] A. Weinberger, "4-2 carry-save adder module," IBM Technical Disclosure Bulletin, vol. 23, no. 8, Jan. 1981, pp. 3811-3814.
- [69] G. Goto et al., "A 4.1-ns compact 54x54-b multiplier utilizing sign-select Booth encoders," JSSC, vol. 32, no. 11, Nov. 1997, pp. 1676-1682.
- [70] J. S. Lim, "Two-Dimensional Signal and Image Processing," Prentice Hall, Sep. 1989.
- [71] IEEE 754 double precision binary floating-point format: binary64, IEEE Standard for Floating-Point Arithmetic (IEEE 754-2008), IEEE, 2008
- [72] Intel Advanced Vector Extensions, see also at <http://software.intel.com/en-us/avx/>
- [73] C6000 Digital Signal Processors, Texas Instruments, see also at http://www.ti.com/lscs/ti/dsp/c6000_dsp/overview.page
- [74] Virtex-6 FPGA Family Overview, Xilinx, DS150 (v2.4), Jan. 2012.
- [75] I. Kuon and J. Rose, "Measuring the Gap between FPGAs and ASICs" IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 2, Feb. 2007, pp. 203 - 215.

- [76] M. Kein, "Power Consumption at 40 and 45 nm," Xilinx White papers, WP298 (v1.0), Apr. 2009
- [77] Virtex-6 FPGA DSP48E1 Slice User Guide, Xilinx, UG369 (v1.3), Feb. 2011
- [78] LogiCORE IP AXI Block RAM (BRAM) Controller v2.00a Product Guide, Xilinx, PG078, Dec. 2012.
- [79] ML605 Reference Design User Guide, Xilinx, UG535 (v1.0), Sep. 2009.
- [80] ISE Design Suite: System Edition, <http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm>
- [81] C. Solomon and T. Breckon, "Fundamentals of digital image processing - A practical approach with examples in MATLAB," John Wiley & Sons, 2011
- [82] Virtex-6 Getting Started Guide, Xilinx, UG533 (v1.4), Nov. 2010
- [83] PicoBlaze 8-bit Embedded Microcontroller User Guide, Xilinx, UG129, June 2011
- [84] TUSB3210, Universal Serial Bus General-Purpose Device Controller, Data Manual, Texas Instruments, Aug. 2007.
- [85] Digital Power Software, Fusion digital power designer, Texas Instruments, http://downloads.ti.com/analog/analog_public_sw/fusion/doc/?DCMP=hpa_pmp_general&HQS=NotApplicable+OT+fusiondocs
- [86] MATLAB User Guide, see the Instrument Control Toolbox, MathWorks web site at <http://www.mathworks.com/products>
- [87] Xilinx Platform Studio, see also <http://www.xilinx.com/tools/xps.htm>
- [88] Xilinx Software Development Kit, Embedded system Tools reference Manual, Xilinx, UG111 (v13.3) Oct. 2011.

- [89] System Generator for DSP, User Guide, Xilinx, UG640 (v13.1), Mar. 2011. see also at <http://www.xilinx.com/tools/sysgen.htm>
- [90] C. Charpentier, “The Simple MicroBlaze Microcontroller Concept,” Xilinx, Application Note: Embedded Processing, XAPP1141 (v3.0), Nov. 2010.
- [91] Digital Clock Manager (DCM) Module, Xilinx LogiCORE, DS485, Apr. 2009.
- [92] IP Processor Block RAM (BRAM) Block (v1.00a), Xilinx, DS444, Mar. 2011.
- [93] LogiCORE IP DSP48 Macro v2.1, Xilinx, DS754, Mar. 2011.
- [94] PG3A Digital Pattern Generator, The Moving Pixel Company, see also at <http://www.movingpixel.com/main.pl?PG3A.html>
- [95] TLA7000 Logic Analyzer, Tektronix, see also at <http://www.tek.com/logic-analyzer/tla7000>
- [96] From a Few Cores to Many: A Tera-scale Computing Research Overview, Intel Research, White Paper, 2006.
- [97] R. Dreslinski et al., “Near Threshold Computing: Overcoming Performance Degradation from Aggressive Voltage Scaling,” Proc. Workshop Energy-Efficient Design, 2009, pp. 44-49.
- [98] D. D. Lu et al., “A Multi-Gate MOSFET Compact Model Featuring Independent-Gate Operation,” IEDM 2007, pp. 565-568.
- [99] P. Kolar et al., “A 32 nm High-k Metal Gate SRAM With Adaptive Dynamic Stability Enhancement for Low-Voltage Operation,” JSSC, vol. 46, no. 1, Jan. 2011, pp. 76-84.
- [100] G. J. Sullivan, “Video Compression - From Concepts to the H.264/AVC Standard,” Proc. of the IEEE, vol. 93, no. 1, Jan. 2005.
- [101] A. K. Jain, “Fundamentals of Digital Image Processing,” Prentice Hall, Oct. 1988.

- [102] W. Heitzelman, "Application-Specific Protocol Architectures for Wireless Networks," Ph. D. Dissertation, MIT, June 2000.
- [103] S. R. Vangal et al., "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," JSSC vol. 43, no. 1, Jan. 2008, pp. 29-41.
- [104] S. Bell et al., "TILE64 processor: A 64-core SoC with mesh interconnect," Digest of Technical Paper, ISSCC, pp. 88-598, Feb. 2008.
- [105] U. Nawathe et al., "Implementation of an 8-core, 64-thread, power-efficient SPARC server on a chip," JSSC, vol. 43, no. 1, 6-20, 2008.
- [106] K. Asanovic et al., "The landscape of parallel computing research: A view from Berkeley," Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, 2006.
- [107] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," Computer, vol. 41, no. 7, 33-38, 2008.
- [108] R. Dreslinski et al., "Nearthreshold computing: Reclaiming moore's law through energy efficient integrated circuits," Proc. IEEE, vol. 98, no. 2, 253-266, Feb. 2010.
- [109] M. Bucci and R. Luzzi, "Design of Testable Random Bit Generators," CHES 2005
- [110] S. Walia, "PRIMETIME Advanced OCV Technology: Easy-to-Adopt, Variation-Aware Timing Analysis for 65-nm and below," Synopsys white paper, Apr. 2009
- [111] TSMC internal document, Nov. 2012